Beyond Visual Line of Sight (BVLOS) Operation of Unmanned Aerial Vehicles (UAVs) for Antarctic Sea Ice Data Collection

Supervisors

Dr. Graeme Woodward (Wireless Research Centre)

Prof. Philippa Martin (Electrical and Computer Engineering)

Assoc. Prof. Wolfgang Rack (Gateway Antarctica)

Campbell Stefan McDiarmid

August 31, 2020

Abstract

The snow radar has recently been developed to non-intrusively measure Antarctic snow depth from an unmanned aerial vehicle (UAV), a vast practical improvement on traditional methods. Improvements in sensing methods is a critical step towards an automated and more effective collection of snow depth measurements, and therefore ice volume by inference. The research focus is to realise the potential of the snow radar by providing an autonomous, reliable and rapid means of surveying an expansive area. UAVs must operate at low-altitudes (5 m - 15 m) to gather accurate snow depth readings. Operational ranges of data collection UAVs are to extended past 10 km, far beyond the visual line of sight (BVLOS). Implementation of a proof-of-concept (PoC) communications architecture was explored for enabling BVLOS data collection missions. A mesh networking protocol called DigiMesh was implemented as a replacement for point-to-point (PtP) telemetry links. This protocol uses IEEE 802.15.4 based media access control layer (MAC) specifications, and a proprietary physical layer (PHY) implementation. Python middleware was written to utilise DigiMesh compatible radios, as they are not directly supported by the open-source UAV ecosystem. Network bottle-necking between ground control station (GCS) and relay UAV was found to be a constraint of the original design. Higher bandwidth radios using IEEE 802.11n PHY/MAC specifications were implemented for this link, with DigiMesh remaining for the inter-UAV network. The physical channel was investigated by simulating the two-ray model. The theoretical maximum range between GCS and relay UAV varied between 2 km to 55 km, depending on the modulation coding scheme (MCS) used. In addition it was shown, that under ideal conditions with a perfectly flat sea ice cover, the spatial position of the relay UAV can be locally optimised with respect to received signal strength. A method for empirically determining channel characteristics with software defined radios (SDRs) is described. An autonomous centre frequency offset (CFO) correction algorithm was implemented within the GNURadio data collection script, improving the quality of channel data. Recommendations were made for future work on a BVLOS data collection system, most notably replacing RF equipment with IEEE 802.11ah compatible hardware. This standard describes sub 1 GHz (S1G) mesh wireless local area networks (WLANs) with much greater data-rates than DigiMesh. For these reasons, IEEE 802.11ah was deemed the most optimal open networking standard for BVLOS data collection missions. Finally, regulatory recommendations are provided for BVLOS UAV operations in Antarctica, including a telemetry data benchmark, and maximum packet loss threshold. This thesis forms the theoretical and practical basis for realistic tests and BVLOS data collection missions in Antarctica.

Contents

Acknowledgements

I am incredibly grateful for all the help and constant guidance from my three wonderful supervisors: Dr. Graeme Woodward, Prof. Philippa Martin, and Assoc. Prof. Wolfgang Rack. I truly couldn't have pushed through this thesis without their ability to talk me through technical problems, referencing woes, thesis structure, and even time management. Not only do they shine as supervisors and academics, but they always seem to bring boundless enthusiasm and lightheartedness to each and every meeting, no matter how much pressure they were under, or I was under – I always left meetings feeling a happier and more confident person.

A massive thanks to Gateway Antarctica for funding this project. They have been the driving force behind this research, providing me with endless support and opportunities to attend amazing events. I hope that Gateway Antarctica finds the research I have been able to carry out on their behalf useful, and that it plays an important role in collecting Antarctic snow and ice data over the 2020/2021 season.

The staff in both the Wireless Research Centre and the Electrical and Computer Engineering department have been a wonderful help to me over the past two years. Special mention to Kelvin Barnsdale in particular, who has gone out of his way on multiple occasions to assist my research and help with testing, despite his busy schedule. And I of course must extend my gratitude to the University of Canterbury for educating me over the past six years, and for providing me with many wonderful experiences during that time.

Finally to my family and friends, I appreciate the constant support and putting up with my erratic working schedule. They have helped me stay happy and positive throughout this endeavour and I am eternally grateful that I have been surrounded with such wonderful people.

To everyone who has been cheering me on and stressing for me throughout the past two years–

Thank you.

Acronyms & Abbreviations

ADC	Analogue to Digital Converter
AP	Access Point
API	Application Programming Interface
ATC	Air Traffic Control
BER	Bit Error Rate
BLER	Block Error Rate
BPSK	Binary Phase Shift Keying
BVLOS	Beyond Visual Line of Sight
BW	Bandwidth
CAA	Civil Aviation Authority
CCC	Christchurch City Council
CFO	Carrier Frequency Offset
CMP	Compatibility
COMNAP	Council of Managers of National Antarctic Programs
CRC	Cyclic Redundancy Check
DEMUX	Demultiplexer
DFT	Discrete Fourier Transform
DOC	Department of Conservation
EIA	Environmental Impact Assessment
ESC	Electronic Speed Controller
EVLOS	Extended Visual Line of Sight

EVM Error Vector Magnitude

FANET Flying Ad-Hoc Network

FEC Forward Error Correction

FHSS Frequency Hopping Spread Spectrum

FIFO First In First Out

FPGA Field Programmable Gate Array

FPV First Person View

FSPL Free-Space Path Loss

GCS Ground Control Station

GFSK Gaussian Frequency Shift Keying

GPS Global Positioning System

I/O Inputs and Outputs

I2C Inter-Integrated Circuit

ID Identifier/Identification

IDFT Inverse Discrete Fourier Transform

IEEE Institute of Electrical and Electronics Engineers

INC Incompatibility

IoT Internet-of-Things

IP Internet Protocol

ISM Industrial, Scientific and Medical

LED Light Emitting Diode

LEN Payload Length

LOS Line of Sight

LTE Long Term Evolution

LUT Look Up Table

MAC Media Access Control

MCS Modulation Coding Scheme

MIMO Multiple Inputs and Multiple-Outputs

MSG Message

MUX Multiplexer

NZ New Zealand

OFDM Orthogonal Frequency Division Multiplexing

PHY Physical Layer

PoC Proof of Concept

PoE Power Over Ethernet

ppm Parts Per Million

PSK Phase Shift Keying

PtP Point to Point

QAM Quadrature Amplitude Modulation

QPSK Quadrature Phase Shift Keying

RC Remote Controller

RF Radio Frequency

ROS Robot Operating System

RPA Remotely Piloted Aircraft

RSSI Received Signal Strength Indicator

RTL Return To Launch

S1G Sub 1 GHz

SDR Software Defined Radio

SEQ Packet Sequence Number

SHA Secure Hash Algorithms

sps Samples Per Second

SSH Secure Shell

STA Station

STX Start of Text

SYS System

TCP Transmission Control Protocol

UC University of Canterbury

UAV Unmanned Aerial Vehicle

UDP User Datagram Protocol

URM2 Ubiquiti Rocket M2

URM9 Ubiquiti Rocket M9

USB Universal Serial Bus

USRP Universal Software Radio Peripheral

VLOS Visual Line of Sight

VTOL Vertical Takeoff and Landing

WAVE Wireless Access in Vehicular Environments

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Network

WRC Wireless Research Centre

XCTU XBee Configuration and Testing Utility

Chapter 1

Introduction

1.1 Motivation

Human induced climate change, often referred to as global warming, is a commonly discussed and ever-present threat to life on Earth [1]. Research from both academia and industry has been conducted over decades to better understand the effects of climate change [2]. One such system critical to the Earth's climate are the thermal cycles that occur in the seas of the Arctic and Southern Ocean. The research presented here contributes to the measurements of sea ice [3].

Due to the important role sea ice plays for global climate, it is a focus of research to better understand multi-decade trends, and how sea ice volume varies over time [4]. Satellites orbiting the earth have been collecting data on ice coverage for years. In theory the data collected can be used to calculate and infer sea ice volume over time, entirely through observations from orbit. However, knowledge of sea ice thickness and therefore volume is still very limited. All sea ice is covered in a layer of snow that varies in thickness, with snow having vastly different physical properties to sea ice. The sensors on-board the satellite are incapable of distinguishing between snow and ice, but when combined with an external snow depth data-set, ice thickness and thus volume can be inferred. It is believed that with enough data collected on snow depth, an analytical model or algorithm can be trained to approximate sea ice volume from satellite data alone. This "training data" has the potential to fully utilise past, present, and future data obtained from satellites.

In recent years snow radar technology has been developed to improve the way in which snow depth data is collected [3, 5, 6]. Traditional measurement methods are slow, labour intensive, and intrusive, whereas the snow radar is sampling at a period of 0.5 s, doing so in a non-intrusive manner. The system can obtain accurate data from 5 m to 15 m above the surface of the snow. Non-intrusive sensing, inherently minimising the cost and risks associated with human labour in Antarctica. Future design iterations of the snow radar will provide improvements to the current form factor, data quality, and sampling period.

These improvements to traditional methods unlock the potential for automation of the data collection process. In fact, this was one of the driving forces behind the development of the snow radar, the potential of leveraging unmanned aerial vehicle (UAV) technology [6]. The combination of non-intrusive sensing and UAVs enable not only the automation of data collection, but additionally the collection of data in areas inaccessible by humans. Sea ice often cannot be safely traversed for snow depth and ice thickness samples to be gathered in an intrusive manner. A recurring example throughout this thesis of such an environment is the Ross Sea, which is typically traversed with an ice breaker. With the aforementioned combination of technology, snow depth data can be collected over such areas.

To summarise, the following is a list of potential benefits associated with utilising the snow radar by combining with UAV technology:

- The inherent risk to human life from the Antarctic environment is reduced.
- Data can be collected in otherwise inaccessible locations.
- Data collection can be automated or controlled remotely.
- Efficiency and cost of data collection is greatly improved.
- Additional remote sensing tasks can be incorporated to data collection missions.
- Rapid realisation of snow/ice volume observations from satellite measurements alone, once sufficient training data has been collected.

1.2 Aim

At the commencement of this thesis in August 2018, the snow radar has been mounted on a large purpose-build multi-rotor UAV to collect trial measurements in Antarctica [5]. The UAV was manually piloted with raw data stored on-board the UAV. Post-processing after missions was carried out to obtain values for snow depth and information about the snow-ice interface.

The aim of this thesis is to leverage UAV and wireless communications technology to maximise the area surveyed with the snow radar in Antarctic data collection missions. This will be accomplished by developing a proof of concept (PoC) that enables data collection UAVs to fly beyond the visual line of sight (BVLOS), with an approach involving a UAV acting as an airborne wireless communications relay. As a result telemetry links between the GCS and each data collector UAV can feasibly be maintained BVLOS. The range of 10 km was set as a goal, as this is approximately where VLOS is lost between the low-altitude UAV and GCS due to the curvature of the Earth. Key improvements will be developed on the PoC system, along with an exhaustive list of recommendations for future improvements to take place after this thesis. Analysis will be undertaken on Antarctica and New Zealand UAV regulations that affect

data collection missions, and recommendations will be made to clarify and better define such regulations/guidelines.

1.3 Outline of Novel Contributions

It quickly became apparent that the largest challenges to be overcome in order to achieve research aim would revolve around: wireless communications, physical attributes of the unique Antarctic environment, and UAV regulations. Typically wireless communications are primarily point-to-point (PtP) between a UAV's telemetry radio and a GCS computer, along with a separate PtP link if manually flown. Research and development projects have been undertaken to drastically improve the flexibility of inter-UAV networks (at the cost of complexity), often referred to as flying ad-hoc networks (FANETs) [7, 8, 9]. However, FANETs often involve UAVs operating within close proximity of one-another (less than 100 m) due to the inherent range limitations of the technology used, such as IEEE 802.11ac/ax. This does not align with this project's ambition to survey areas of sea ice up to 10 km away from the launch zone. For these reasons, the thesis has been divided into four novel work chapters, each summarised as follows:

- **Proof-of-Concept BVLOS System Design**
  Develop a PoC BVLOS network implementation, defining an engineering benchmark. This will involve one GCS maintaining a telemetry connection with one data collection UAV via a relay UAV.

- **Physical & Spatial Analysis**
  An investigation into the application of a theoretical channel model, and the implications of such a model on maximum operational range. Topics include: physical attributes associated with Antarctica, the spatial implications of using directional antennas, simulations of the two-ray channel model, UAV spatial position optimisation, and how to empirically measure channel characteristics.

- **Final Implementation of BVLOS System & Engineering Recommendations**
  Identify weaknesses of the PoC design and consequently implement improvements. Comprehensive future work section that discusses: future improvements to the wireless communications architecture, beneficial features recommendations, and new research targets.

- **Policy & Regulatory Recommendations Surrounding BVLOS Flight**
  Identifying constraints, ambiguity, and issues within New Zealand and Antarctica UAV regulations, and recommending changes that may affect this project and similar endeavours. Recommendations are specifically made for BVLOS operations and rules that apply to the three main modes of flight: manually piloted, way-point missions, and autonomous flight.

Chapter 2

Background

2.1 Antarctica

The background information chapter will begins with information about Antarctic considerations, operations and relevant information on the snow radar. It is important to outline any and all operational constraints that BVLOS data collection missions may be subject to.

2.1.1 Safety, Operations & Conservation Policy

Antarctica has no governing body, and therefore has no common set of laws that are unique to the continent. In 1959 the Antarctic Treaty was signed by the 12 countries with ongoing scientific operations in Antarctica over the 1957/1958 international geophysical year [10]. Today 54 countries now act in accordance to the treaty for operations in Antarctica. The Secretariat of the Antarctic Treaty maintains the Antarctic treaty, subsequent amendments, and organises meetings between countries that have an active presence in Antarctica. Annual meetings take place between party representatives to discuss recommendations for the operations of all parties on the continent. Outside of guidelines and recommendations within the Antarctic Treaty, individual nations often specify their own set of rules and regulations that specifically apply to their own operations. If something is undefined in the Antarctica Treaty, not specified in any recommendation documents, and also not defined within a nations rules and regulations, then permissive law is assumed. Therefore, in the case of an action or operation that has not being prohibited or regulated by any means, it is assumed to be permitted.

The majority of considerations made will be in accordance with the rules and regulations specified by Antarctica New Zealand (AntarcticaNZ), the crown entity responsible for managing New Zealand's scientific operations and programs [11]. Prominent bases of operations established by New Zealand, such as Scott Base, are managed by AntarcticaNZ. This includes maintaining and upgrading the buildings, and re-supplying Scott Base with the resources required to operate. AntarcticaNZ has defined a set of guidelines for general operation, which should be taken into consideration when practically carrying out BVLOS missions.

Looking specifically at generic aircraft and remotely piloted aircraft (RPA) operations, The Secretariat of the Antarctic Treaty has defined a set of guidelines for each. The following are summarised considerations that are relevant to Ross Sea BVLOS data collection missions, obtained from Environmental Guidelines for operation of Remotely Piloted Aircraft Systems (RPAs) in Antarctica [12]:

- UAV flying, takeoff or landing is prohibited from disturbing wildlife, such as concentrations of seals or birds. Flights should be planned accordingly to avoid such disturbances.
- Flight paths should be pre-planned, and alternative landing zones should be identified in the case of imminent failure.
- UAV operations should consist of a pilot and at least one observer. Note: It is not explicitly stated whether this is this case for single UAV operations, or whether one pilot and one observer is sufficient for multiple UAV operations.
- The cumulative environmental impact should be considered when flying multiple UAVs in a single mission.
- It is implied that global positioning system (GPS) location (or equivalent measure of position) of all UAVs is required, as the last recorded GPS location must be noted in the case of a crash.
- BVLOS missions must be approved by a competent authority, and where possible an observer maintaining a VLOS should monitor wildlife activity and inform the pilot of potential behavioural changes.

In addition, the Council of Managers of National Antarctic Programs (COMNAP) has defined a UAV operators handbook [13]. This contains relevant definitions, a comprehensive flowchart on pre-flight planning, a guide on risk assessment and flight record keeping. The following is a summary of the COMNAP pre-flight plan relevant to BVLOS flight operations over the Ross Sea:

1. UAV activity: Scientific Operations.
2. Vehicle size class: 2 kg - 25 kg.
3. Perform Environmental Impact Assessment (EIA).
4. Other air operations taking place in area: No.
5. Adhere to COMNAP rules/restrictions and any national legislation.
6. Carry out risk assessment and proceed to fly if the outcome is satisfactory.

7. Save flight logs for sending to COMNAP, as they intend on keeping a record of any and all UAV operations in Antarctica.

Despite all of the above mentioned considerations, there appears to be no publicly radio spectrum management policy defined by The Secretariat of the Antarctic Treaty, AntarcticaNZ or COMNAP. Therefore, the wireless communications of BVLOS data collection system will be developed under the assumption that internationally recognised industrial, scientific and medical (ISM) frequency bands will be acceptable. The relevant frequency bands that will be investigated are the 433 MHz, 900 MHz and 2.4 GHz bands.

2.1.2 Environmental Conditions

Antarctica experiences some of the most extreme weather conditions on the planet, and these differ significantly between seasons. BVLOS data collection missions can only feasibly take place during the summer season. Therefore, a breakdown of unique Antarctic environmental conditions will be provided, with each of these aspects being investigated for potential implications on wireless communications and UAV operations.

While it would be interesting to isolate the effects of the weather conditions on the communications system, it is likely that weather will make an unavoidable contribution to the results of other testing such as channel sounding. Nothing can be done about this, so at this stage, so while testing is being carried out the best course of action will be to collect weather information via on-board sensors or weather forecast, then analyse results after the data has returned from Antarctica. It has been found that as temperature decreases, RSSI increases and the noise floor decreases [14]. Therefore, the BVLOS communications system will perform better in Antarctic temperatures, assuming control for all other factors.

Another notable consideration will be the electromagnetic reflectivity of snow and sea ice. This is important to consider, due to a phenomena called the two-ray ground reflection model, and will be covered in Section 2.3.2. The strength of the reflected signal depends on a variety of factors, the most important being the relative permittivity, ϵ_r of the reflective surface [15]. For BVLOS data collection missions over the Ross Sea, ϵ_r will be attributed to the snow covered sea ice. Measurements and studies have found value of ϵ_r to be 3.17 ± 0.07 for naturally occurring snow-covered sea ice [16], and 3.185 for dry snow [17]. Relative permittivity is a frequency dependant property. However, the above values remain constant for the ISM frequency bands mentioned, of 433 MHz, 900 MHz, and 2.4 GHz.

2.1.3 Snow Radar

Data collection missions are primarily constrained by the sensory payload, the snow radar. This device is responsible for collecting snow thickness data, which combined with satellite measurements can be used to infer the thickness of sea ice. Furthermore, when combined with

satellite imagery, the volume of both snow and ice can be inferred over a surveyed region.

The first working design of the snow radar has been published [3], containing key parameters for which BVLOS data collection missions will be based around. For accuracy purposes, the maximum operational height specified is 5.0 m above the snow surface. This design iteration of the snow radar has an operational frequency of 1.5 GHz - 4.5 GHz, physical dimensions of 40 x 40 x 20 cm, and a mass of 3.22 kg. Due to the relatively large physical dimensions and mass of the snow radar, this design iteration can only be mounted onto powerful multi-rotor UAVs for extremely low altitude operations. This was done so for design validation during the 2016-2017 summer, where the snow radar was mounted on a multi-rotor UAV, operating at an altitude of 5.0 m and moving at a velocity of 2.0 m/s during data collection [5]. When standard multi-rotor UAVs move with a constant velocity while maintaining a constant altitude, the UAV will physically tilt forwards in the direction of movement. Therefore, the physical constraints that eliminate the potential use of fixed-wing UAVs, inadvertently introduce another factor to be accounted for in tilt angle. This had to be taken into consideration when processing the sensor data, although a gimbal control system could be implemented to mitigate the effect of tilt on the raw data.

Future design iterations of the snow radar are underway, with the aim to reduce the size and mass of the sensor, while increasing sampling rate and the altitude constraint. For the purpose of this project, BVLOS mission benchmarks will take into account the constraints of the current snow radar design. However, the design improvements that come with future iterations of the snow radar will inevitably result in less strict constraints. For example, a physically smaller and lighter snow radar that can collect data faster and at higher altitudes could be equipped to a fixed-wing UAV. Therefore, both multi-rotor and fixed-wing UAV frames will be taken into account for design considerations of the BVLOS UAV system.

2.2 Modern Unmanned Aerial Vehicles

Leverage unmanned aerial vehicle (UAV) technology is key to autonomously collecting snow radar data over expansive areas. This section will cover the essential information about the standards, protocols, hardware and software of modern UAV technology.

2.2.1 MAVLink Protocol

The Micro Air Vehicle Link (MAVLink) protocol was released by Lorenz Meier in 2009 as part of an ambitious project to create the first autonomous drone with flights navigated predominantly by computer vision [18]. This same project also paved the way for the first generation Pixhawk flight controller and the cross-platform ground control station (GCS) known as QGroundControl [19]. MAVLink has since become a fundamental protocol for communicating with drone flight controllers and on-board peripheral devices such as GPS, cameras, sensors, and electronic

speed controllers (ESCs), using a common set of message definitions.

MAVLink serves as a middleware protocol to simplify data transmission between the flight controller, UAV peripherals, GCS and sometimes a companion computer. The following list describes common uses of the MAVLink protocol for modern UAV operations:

- GCS dashboards contain information from the flight controller and an interface to carry out important actions with MAVLink, such as:
  - Flight controller sensor and peripheral calibration.
  - Configuring remote controller (RC) action bindings for different buttons and switch states.
  - Reading and writing flight controller parameter values using parameter requests.
  - Periodic feedback of flight data such as GPS coordinates, altitude, tilt, and yaw; all of which are communicated over a telemetry radio link.
  - Loading a set of way-points for an autonomous mission onto the flight controller's memory.
- RC displays show MAVLink feedback for critical flight data such as battery levels, altitude, tilt, and yaw.
- Wired connections between a flight controller and an on-board computing device (companion computer) use MAVLink for advanced UAV operations, such as collision avoidance and Robot Operating System (ROS) interfacing.

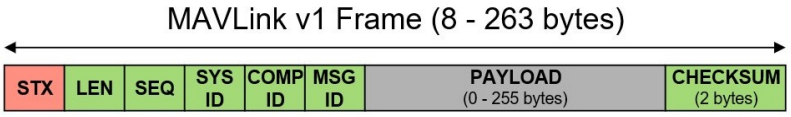


Figure 2.1: Graphical representation of the MAVLink v1 packet structure [20].

MAVLink v1 frames have a six byte header, a payload of up to 255 bytes followed by a two byte X25 cyclic redundancy check (CRC) checksum, as shown in Figure 2.1. Each of the header bytes of the MAVLink v1 frame are designated for the following purposes:

0. Start of text (STX) byte indicating MAVLink protocol version (0xFE for v1, 0xFD for v2).
1. Payload length (LEN) in bytes (0-255).
2. Packet sequence number (SEQ) is a counter incremented after each message and rolls back over to 0 after 255. This is used as a way to detect packet loss in telemetry links (0-255).

3. System identifier (SYS ID) is a uniquely assigned value to distinguish between different vehicles (1-255). GCS uses the reserved broadcast value 0.

4. Component identifier (COMP ID) refers to the component sending the MAVLink message. For example, a flight controller and MAVLink compatible camera on the same vehicle will both have a different COMP ID and the same SYS ID (1-255).

5. Message identifier (MSG ID) denotes the type of MAVLink message that has been transmitted, enabling the receiving device to look-up how to interpret the payload (0-255). This means that information about the message structure does not have to be included within the payload and this therefore drastically lowers the overhead on each frame.

Figure 2.2: Graphical representation of the MAVLink v2 packet structure, with an additional header byte each for both incompatibility and compatibility flags, two additional bytes for MSG ID and optional packet signing [20].

MAVLink v2 has extended the header from six to 10 bytes by adding two new fields and extending the MSG ID field from one to three bytes, while also making improvements to efficiency by truncating any zero-filled bytes from the end of payloads. At the cost of four bytes of overhead, these changes allow for more developer features and over 16 million message definitions up from 255. An optional 13 byte message signature was also added in MAVLink v2 to improve the security of the connection between GCS and UAV by mitigating the risk of a third party GCS seizing control. A breakdown on the signature contents can be seen in Table 2.1.

MAVLink v2 Signature		
Field	**Bits**	**Description**
Link ID	8	Unique drone and channel ID to distinguish between different MAVLink systems in a network.
Timestamp	48	Time elapsed since 1st January 2015, with 10 μs precision.
Signature Hash	48	First 48 bits of a SHA-256 hash of the entire packet along with the link ID and timestamp portion of the signature. A 32-bit secret key is manually defined at both ends.

Table 2.1: Breakdown of the MAVLink v2 optional signature [21].

The two new header fields seen in Figure 2.2 are designated for incompatibility flags (INC FLAGS) and compatibility flags (CMP FLAGS). Incompatibility flags must be understood by the receiving device's MAVLink library in order for the message to be handled – if an unknown incompatibility flag is raised then the message is discarded by the receiver. By default, the only supported flag is bit zero, used to indicate whether message signing is being used and that a signature at the end of the frame is present. Compatibility flags were introduced for developers to indicate non-critical features of MAVLink messages. There are no default compatibility flag cases, the field is there for developers to introduce new functionality to their UAV systems. An example from the MAVLink website on use cases for the compatibility flag field would be to indicate message priority [20]. This is an example of when to use compatibility flags over incompatibility flags, because a GCS using default MAVLink would have no definition for prioritisation flags. If incompatibility flags were used to indicate heightened priority the message would be discarded by a GCS with the default MAVLink v2 library, whereas if compatibility flags is used the message will merely be handled without prioritisation – a far more favourable worst case scenario.

Version 2 of the MAVLink protocol is the most up-to-date version of the protocol, supported by all major open-source flight controller firmware stacks and will be forward compatible with devices running future versions of MAVLink [18]. At this stage no future versions of the protocol will be released any time soon, although MAVLink author Lorenz Meier has discussed what the next iteration of the MAVLink protocol could introduce when it is eventually released at the PX4 Developer Conference 2019. Some notable features discussed during the talk are: improved security, revised data-types, increased maximum payload size, guaranteed message delivery and message routing in IP networks [22].

2.2.2 UAV Technology Stack

With the exclusion of MAVLink that was discussed in greater depth, this subsection provides an overview of the modern UAV technology stack This includes an explanation of the purpose of each item, along with any specific variant that was used throughout the thesis. Various alternatives to UAV technology have been summarised throughout Appendix A.

- **Flight Controller**
  At the heart of every modern UAV is a flight controller capable of running autopilot software with enough inputs and outputs (I/O) to connect the following peripherals: telemetry radio, RC radio, ESC, GPS receiver, power unit, servos, additional sensors. A summary of options for flight controller hardware can be found in Table A.2. The Pixhawk 1 flight controller has been used throughout this thesis due to high performance, large user-base, and availability.

- **Autopilot**
  The autopilot is the software stack operating on the flight controller hardware, exchanging data between peripherals, external computing devices, and the GCS using MAVLink protocol. A summary of options for autopilot software can be found in Table A.1. PX4 has been used throughout this thesis due to the large user-base, rich features, and was originally designed for Pixhawk hardware.

- **Ground Control Station**
  The GCS application provides advanced control and telemetry feedback of a UAV from an often stationary ground position. Features of the GCS include: providing firmware updates to hardware components, sensor calibration, reading/writing firmware settings, and planning missions with way-point maps. A summary of options for GCS software and compatible operating systems can be found in Table A.3. For the purpose of consistency and cross-platform compatibility, QGroundControl has been the only GCS application used throughout this thesis.

- **Telemetry Radio**
  Telemetry is the real-time feedback of sensory and state information from the UAV. The purpose of the telemetry radio is to maintain a reliable link between UAV and GCS for the transmission of such information and receiving of mission map data. Typically telemetry radios utilise SiK firmware that strictly supports a point-to-point (PtP) network topology, and operates on 433 MHz or 915 MHz frequencies.

- **Companion Computer**
  The companion computer is a non-essential component in the modern UAV technology stack, with the purpose of performing real-time computing tasks to assist the autopilot. It is most commonly used for integrating autonomy into flight with ROS, but can also be used to connect peripherals and sensors to the UAV that are otherwise incompatible with the flight controller. Any computer with a small form factor can realistically be used as a companion computer, e.g. a Raspberry Pi. Each of the NAVI UAVs available are equipped with an Intel NUC5i7RYH running a Linux based operating system, and is the only variant of companion computer used throughout this thesis.

- **Air-frame**
  Multi-rotor air-frames have been uses exclusively throughout this thesis, and mention has been made of fixed with UAVs for their performance and potential future in snow radar data collection.

- **Modes of Flight**
  Strictly speaking, two modes of flight exist for UAVs with modern flight controllers and autopilots: real-time directional vector commands, and way-point destination commands. Directional vector commands can be associated with manual flight, where a pilot controls a UAV in real-time with a RC. Way-point destination commands can be associated with

mission maps designed within a GCS application, all movements are determined by the autopilot. Autonomy could be considered a third mode of flight, as the UAV can make decisions on-board based on sensory information and software. However, the commands passed from autonomous decision making software to the UAV will either be in the form of a movement vector, or a way-point.

2.2.3 New Zealand UAV Policy

Civil Aviation Authority (CAA) states that visual line of sight (VLOS) requires [23]:

> For the purposes of this rule visual line of sight means a straight line along which an observer has a clear view and which may be achieved with the use of—
>
> 1. spectacles, contact lenses, or a similar device used to correct subnormal vision of the user to no better than normal vision but not the use of an electronic, mechanical, electromagnetic, optical, or electro-optical instrument; or
> 2. a first person view system and a trained and competent observer who maintains—
>   (a) visual line of sight of the aircraft; and
>   (b) sight of the surrounding airspace in which the aircraft is operating; and
>   (c) direct communication with the person who is operating the aircraft.

While this project is aiming to design an Antarctic capable UAV system, all field testing will be carried out in New Zealand prior to deployment in Antarctica. Therefore, this project must also take the current state New Zealand UAV regulations into close consideration, so rules and definitions from the CAA will be used as a guide. "The Civil Aviation Authority is a Crown entity responsible to the Minister of Transport. Civil aviation in New Zealand operates within a system established and maintained by the Civil Aviation Act 1990 [24]."

Part 101, sub-part A (101-A) specifies that all UAV operators intending to fly within a controlled airspace must obtain prior approval from Air Traffic Control (ATC) unless the operation is shielded. Part 101-E requires VLOS at all times while carrying out a UAV operation, implying that the pilot must have an unobstructed view of the UAV and surrounding airspace. Additionally, a direct communications link must also be present between the UAV and operator [23]. Due to the ambiguity of this rule, it is unclear whether operating a UAV via relayed communications link follows CAA guidelines. Despite all of the rules laid out in Part 101, there is no distinction between manual flight, programmed flight paths or autonomous flight. Part 102 describes guidelines for awarding UAV pilots with an unmanned aircraft operator certificate. Holders of this certificate are required to comply with Part 101, but not required to comply with various other Parts of CAA's rules and regulations[1][25].

[1]Exempt from Parts 12, 19, 21, 26, 39, 43, 47, 61, 63, 66, 67, 91, 92, 93, 95, 115, 119, 129, 133, and 137.

Part 71 provides descriptions for different classifications of airspace, and guidelines for any required communication with ATC [26]. In this project all small-scale tests will take place in a shielded zone, and large-scale tests inside a restricted airspace that is activated by contacting ATC. Shielded operations in a controlled airspace are subject to the restrictions specified in Part 101 and 102. The restricted airspace classification has been given to the University of Canterbury's test range at Birdling's Flat, a 10 km by 10 km section spanning over land and water. By default is not "active", and prior to flight tests in this area the pilot must request for ATC to activate the airspace, in turn allowing for flight of up to a 1 km altitude. Such flight is exempt from restrictions described in Part 101 and 102 at the discretion of the pilot. It is worth noting that Part 71 was last amended in October 2008 [26].

2.3 Wireless Channel/Physical Medium

2.3.1 Free-Space Path Loss

A link budget analysis is used to determine the received signal power, assuming a clear line of sight and no distortion from other transmitters or reflections [27] [28]

$$P_r = P_t\left[\frac{\sqrt{G_l}\lambda}{4\pi d}\right]^2 \tag{2.1}$$

where P_r is received signal power, P_t is transmitted power, G_l is the net gain of transmitter and receiver hardware (antennas and cables), and λ is wavelength. Equation 2.1 is often represented in logarithmic form and decibel values for antenna gain, transmitter power and receiver sensitivity.

2.3.2 Two-Ray Model

The two-ray ground reflection channel model features throughout Chapter 4, as the Ross Sea channel is anticipated to be resemble the model.

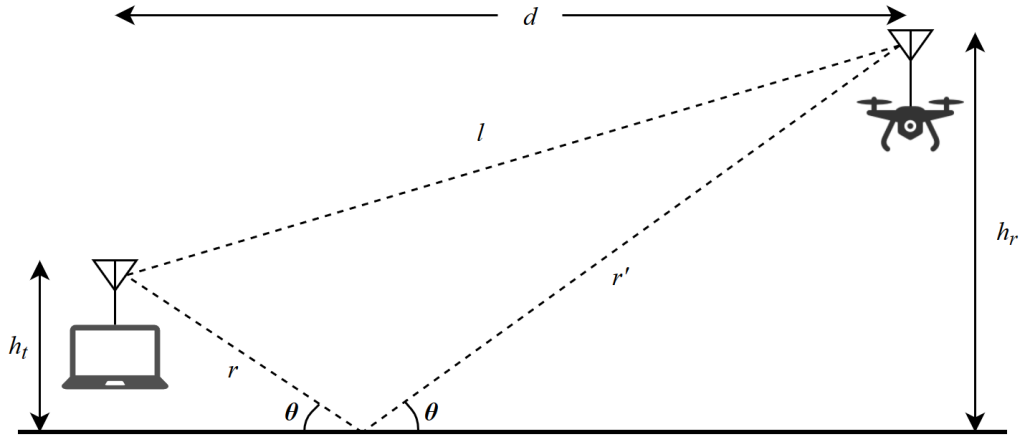


Figure 2.3: Generic diagram describing the two-ray channel model and illustrating where parameters align with the relevant equations.

Two primary paths, the line of sight (LOS), l, and non-LOS path, $r+r'$, have been illustrated in Figure 2.3, along with other spatial parameters. These parameters include transmitter height, h_t, receiver height, h_r, angle of incidence, θ, and horizontal distance, d. The received power

$$P_r = P_t\left[\frac{\lambda}{4\pi}\right]^2\left|\frac{\sqrt{G_l}}{l} + \frac{\Psi\sqrt{G_r}\exp\left(j\Delta\phi\right)}{r+r'}\right|^2, \tag{2.2}$$

can be calculated with the spacial parameters shown in Figure 2.3. Values for l, $r+r'$, θ and $\Delta\phi$ can be calculated given h_r, h_t and d. The reflection coefficient

$$\Psi = \frac{\sin\theta - Z}{\sin\theta + Z}, \tag{2.3}$$

is a function of θ and

$$Z = \begin{cases} \sqrt{\epsilon_r - \cos^2\theta}/\epsilon_r & \text{for vertical polarisation} \\ \sqrt{\epsilon_r - \cos^2\theta} & \text{for horizontal polarisation} \end{cases}, \tag{2.4}$$

where Z is a function of the relative permittivity of the reflective surface, ϵ_r and θ.

2.3.3 Fresnel & Diffraction Models

Building upon the idea of multi-path reflection, Fresnel Zones are an elliptical model of the communications channel and all obstructions that can potentially cause a reflected or diffracted path between transmitter and receiver [15]. This model specifically helps identify the phase

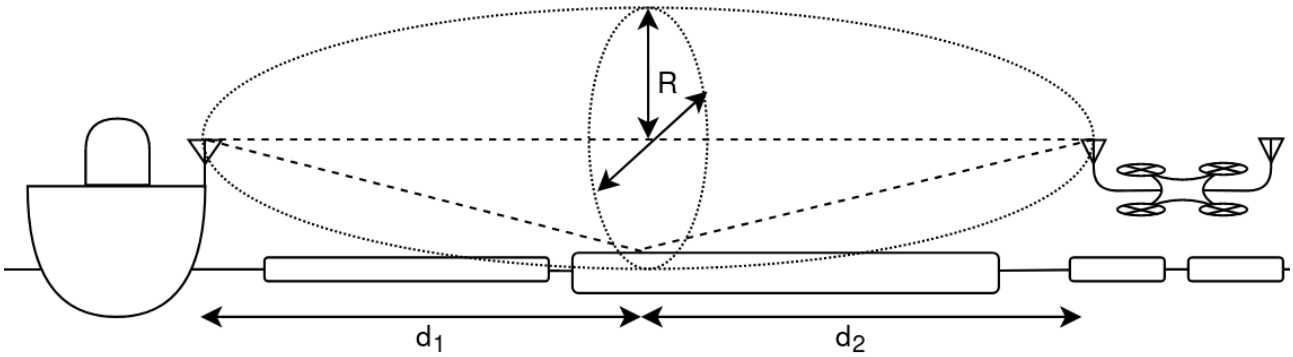


Figure 2.4: Basic illustration of the first Fresnel Zone in relation to the primary point of reflection.

shift of reflected signals, and thus determine whether phase cancellation will occur [29]

$$R = \sqrt{\frac{\lambda d_1 d_2}{d_1 + d_2}} \tag{2.5}$$

where λ is wavelength, d_1 and d_2 are defined Figure 2.4. As blockage of the Fresnel zone increases, the quality of the received signal diminishes and therefore loss of signal can occur [30]. If the distance between the Fresnel Zone and Earth, known as clearance, is greater than 60% of the radius of the Fresnel Zone, then communications are considered to be free from the adverse effects of diffraction and obstruction [31].

2.4 Relevant Protocol Sets & Technologies

2.4.1 DigiMesh

Specifically, the Digi XBee S3B 900HP Pro, with DigiMesh firmware (version number 8075), will be used in the development of UAV networks for BVLOS operations. DigiMesh is a self-forming, self-healing, and automatic routing mesh networking protocol based off IEEE 802.15.4 media access control layer (MAC), and uses a proprietary physical layer (PHY). This PHY involves data being modulated using Gaussian frequency shift keying (GFSK), and transmitted using frequency hopping spread spectrum (FHSS). DigiMesh has a network capacity of 91 kbps with no encryption with one network hop [32].

2.4.2 Relevant IEEE 802.11 Standards

Each OFDM sub-carrier uses a digital modulation technique and coding rate depending on the modulation coding scheme (MCS) selected. The generic modulation techniques used are phase-shift keying (PSK), and quadrature amplitude modulation (QAM). More specifically, 802.11 uses binary PSK (BPSK), quadrature PSK (QPSK), 16-QAM, 64-QAM, and 256-QAM.

Wireless access in vehicular environments (WAVE) is described in 802.11p-2010 (amendment 6) [33]. Hardware supporting this standard will be able to account for a constantly changing channel and the Doppler-shifting for high-velocity UAVs. While the target application of this standard aligns with inter-UAV missions, the frequency band intended for use is 5.85-5.925 GHz. The 802.11p standard is intended for high data-rate, short range communications between vehicles and nearby urban infrastructure. Therefore, 802.11p is not appropriate for BVLOS Antarctic data collection missions.

Mesh networking was officially introduced in 802.11s-2011 (amendment 10) [34]. The mesh networking capabilities introduced in 11s have been since integrated into the 802.11ac/ax standards, widely used for simple home/office mesh networks.

802.11n was an amendment to the 802.11 family of standards that offered improvements to network speed through a means of larger bandwidths and utilising multiple inputs and multiple outputs (MIMO) technology [35]. This is used throughout this research, as Ubiquiti Rocket M2 hardware, used in the later stages of system development, utilises this networking standard.

The 802.11ah-2016 (amendment 2) standard introduced PHY and MAC specifications for sub 1 GHz (S1G) implementations of a 802.11 wireless local area network (WLAN) [36]. Included in this standard is mandatory support for 1 and 2 MHz channel widths, and optional support for 4, 8, and 16 MHz channel widths. MCS with similar parameters to those described in Table 2.3 are also available, data-rates for each MCS can be calculated using Equation 2.6. This would

provide a variety of options for choosing the optimal MCS for the desired range. Additionally, this standard supports mesh networking that was described in 802.11s, along with mitigation of the Doppler-shifting effect mentioned in 802.11p. Both mesh and non-mesh S1G stations (STAs), along with relay access points (APs).

Table of Symbols

Symbol	Description
B	Channel Bandwidth.
N_{SP}	Pilot sub-carriers per channel. 802.11n uses $N_{SP} = 4, 6$, for $B = 20, 40$ MHz. 802.11ah uses $N_{SP} = 2, 4, 6, 8, 16$, for $B = 1, 2, 4, 8, 16$ MHz respectively.
N_{SD}	Complex data numbers per spatial stream per OFDM symbol. 802.11n uses $N_{SD} = 52, 108$ for $B = 20, 40$ MHz respectively. 802.11ah uses $N_{SD} = 24, 52, 108, 234, 486$ for $B = 1, 2, 4, 8, 16$ MHz respectively.
N_{ST}	Non-null sub-carriers per channel. The outer-most sub-carriers in each channel overlap heavily with orthogonal channels, and therefore remain null. $N_{ST} = N_{SP} + N_{SD}$.
N_{SS}	Spatial streams.
R	Coding rate. The proportion of data to overhead in a coding scheme.
N_{BPSC}	Coded bits per single carrier. Varies based on the modulation method used in the MCS, where $2^{N_{BPSC}}$ is equal the number of points in the constellation. $N_{BPSC} = 1, 2, 4, 6, 8$ for BPSK, QPSK, 16-QAM, 64-QAM, and 256-QAM respectively.
Δ_F	OFDM sub-carrier frequency spacing. $\Delta_F = 312500$ Hz for both 20 MHz and 40 MHz BW settings in 802.11n. $\Delta_F = 31250$ Hz for all B in 802.11ah.
T_{DFT}	Discrete fourier transform (DFT) and inverse discrete fourier transform (IDFT) time period. $T_{DFT} = 1/\Delta_F$.
T_{GI}	Guard interval. This takes place after transmissions to minimise the effect of multi-path propagation at the receiving end. 802.11 OFDM originally specified this to be $T_{GI} = T_{DFT}/4$, although 802.11n introduced optional support for a shorter guard interval $T_{GI2} = T_{DFT}/8$.

Table 2.2: Summary of symbols used in data-rate and error calculations for 802.11.

Relevant 802.11n MCS			
MCS	**Modulation**	**Code Rate**	**EVM**
0	BPSK	1/2	-5 dB
1	QPSK	1/2	-10 dB
2	QPSK	3/4	-13 dB
3	16-QAM	1/2	-16 dB
4	16-QAM	3/4	-19 dB
5	64-QAM	2/3	-22 dB
6	64-QAM	3/4	-25 dB
7	64-QAM	5/6	-27 dB

Table 2.3: Summary of different MCS as specified by 802.11 standards [35]. MCS 8-15 are identical with respect to modulation, code rate and EVM, the only difference lying in N_{SS}.

A summary of important physical parameters used for calculating network capacity in 802.11 standards is shown in Table 2.2. Each MCS has a unique set of parameters for modulation method, N_{BPSC}, and coding rate, R. Increasing network speed/capacity results in a lower error vector magnitude (EVM), as can be seen in Tables 2.3 and 2.4. EVM can be likened to a gain when estimating the maximum range of a MCS. The difference between MCS 0 and MCS 7 of 22 dB indicates that MCS 0 can tolerate an additional 22 dB of attenuation or path loss.

Extra MCS Supported by 802.11ah			
MCS	**Modulation**	**Code Rate**	**EVM**
8	256-QAM	3/4	-30 dB
9	256-QAM	5/6	-32 dB
10	BPSK×2	1/2	-4 dB

Table 2.4: Summary of additional MCS specified in the 802.11ah standard, MCS 0-7 modulation, code rate and EVM are the the same as those described in Table 2.3 [36]. MCS 10 invokes 2× repetition, and is only available for $B = 1$ MHz, $N_{SS} = 1$.

The data-rate for a given 802.11n/ah MCS, D_{MCS}, can be calculated as

$$D_{MCS} = \frac{N_{SD} N_{BPSC} N_{SS} R}{T_{GI} + T_{DFT}}, \tag{2.6}$$

where Table 2.2 provides a definition for each of the parameters used in the calculation.

Chapter 3

Prototype – Relayed Control of a UAV

This chapter will cover the design for a proof-of-concept system for BVLOS control of UAVs. Modular design philosophy enabled rapid development of later iterations of the system with optimisations made. The following will be covered in depth: the concept of a communications relay; replacing the SiK radio with hardware capable of supporting mesh networks; modular design overview and implementation of the new radio; an in-depth analysis of the software designed; and discussion of the positives and drawbacks of this proof-of-concept design.

3.1 Concept – Communications Relay UAV

Beyond visual line of sight (BVLOS) UAV operation poses both a logistical challenge from a pilot's perspective, and an engineering challenge from a wireless communications perspective. Both of these challenges are addressed in this chapter, with more emphasis placed on the wireless communications challenge. Standard GCS and UAV communications use point-to-point connections either directly through a telemetry radio link (433MHz, 900MHz or 2.4GHz) or over the internet via a WiFi network (2.4GHz or 5GHz). The MAVLink protocol also supports connections over a cellular network, assuming the UAV has been configured to use an on-board 4G or LTE transceiver in place of the telemetry radio. This offers significant improvements to the range of operation in urban areas at the cost of added latency. However, this is not an option in an environment devoid of cellular infrastructure such as Antarctica's Ross Sea, where the BVLOS missions will be carried out. Therefore, a non-standard approach must be taken to maintain a reliable telemetry link with all data collection UAVs, and hence, the decision to incorporate an airborne communications relay UAV into missions.

We now identify, illustrated in Figure 3.1, the key engineering challenges for an airborne relay scenario. Both ArduPilot and PX4 are only compatible with telemetry radios that are running SiK, an open source telemetry radio firmware for direct point-to-point communications. A pair of SiK telemetry radios are required for each standard telemetry link. For example, if n UAVs are to be wirelessly communicating with one GCS, n radios would be plugged into the GCS computer to wirelessly communicate with each of the n UAVs, bringing the total number of

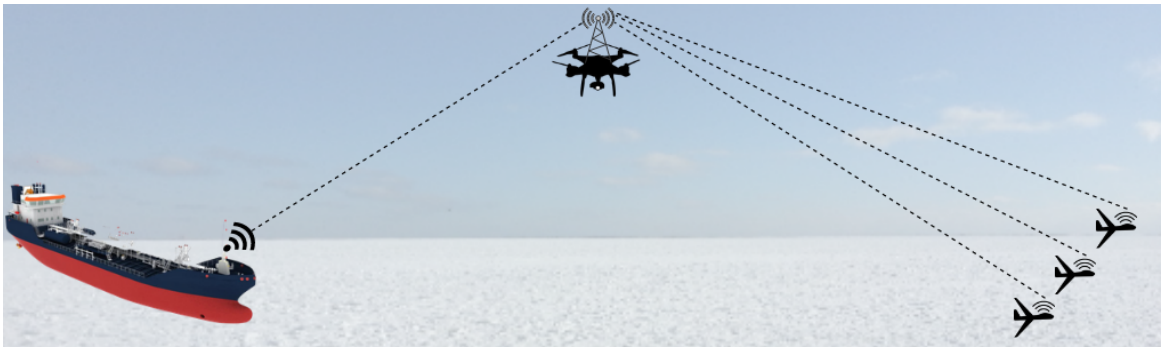


Figure 3.1: Illustration of the conceptual solution to BVLOS communication between a GCS and three low-altitude data collector UAVs via a relay UAV.

SiK radios to $2n$. If each of these UAVs is operating BVLOS, a communications relay UAV is required to maintain telemetry links. This adds another radio to the GCS ($n+1$ radios) and the relay UAV has its own telemetry link plus a pair of radios for each data collector ($2n+1$ radios), bringing the total number SiK radios to $4n + 2$. This is illustrated in the first row of Figure 3.2, showing that exclusively using point-to-point radio links for BVLOS operations involving multiple UAVs is not scalable in terms of hardware requirements and radio spectrum allocation.

It became apparent that either a mesh network and/or hybrid network would be required for a suitable and scalable replacement to the SiK telemetry radio's point-to-point topology. IEEE 802.15.4 defines a set of MAC and PHY layer standards for wireless personal area networks (WPANs) constrained to low-bandwidths (up to 250 kbit/s), and is often used in internet of things (IoT) devices or sensor networks [37]. These standards are the basis for mesh networking protocol stacks such as ZigBee, Thread, and DigiMesh (based on ZigBee) [38, 39, 40]. The latter is featured prominently in Figure 3.2 as compatible 915 MHz radio hardware was readily available for development. Each node in a DigiMesh network acts as a router/coordinator, rather than each node having a predetermined role such as router, coordinator or endpoint. These features, along with dynamic network discovery and routing mean that a highly scalable, low bandwidth network can be deployed with minimal configuration.

WiFi is based on the IEEE 802.11 family of standards, and networks use a star topology to connect multiple endpoint devices to one coordinator, which is depicted in rows 4, 5 and 6 of the topology illustrations in Figure 3.2. Traditionally WiFi is used to connect a large number of endpoint devices to one coordinator/router in a star topology, requiring endpoints to be within a close proximity to the coordinator/router [35]. High-power WiFi modems have been developed for longer-range connections between nodes, enabling the possibility of incorporating higher bandwidth connections into BVLOS UAV operations. However, even high-power 2.4 GHz transceivers with a WiFi network topology will limit the range of operations when compared to network implementation using lower frequency (433 MHz or 915 MHz) transceivers. The implementation of a WiFi only topology also implies that the relay UAV must act as the coordinator node due to WiFi not supporting multiple routers or coordinators in the same

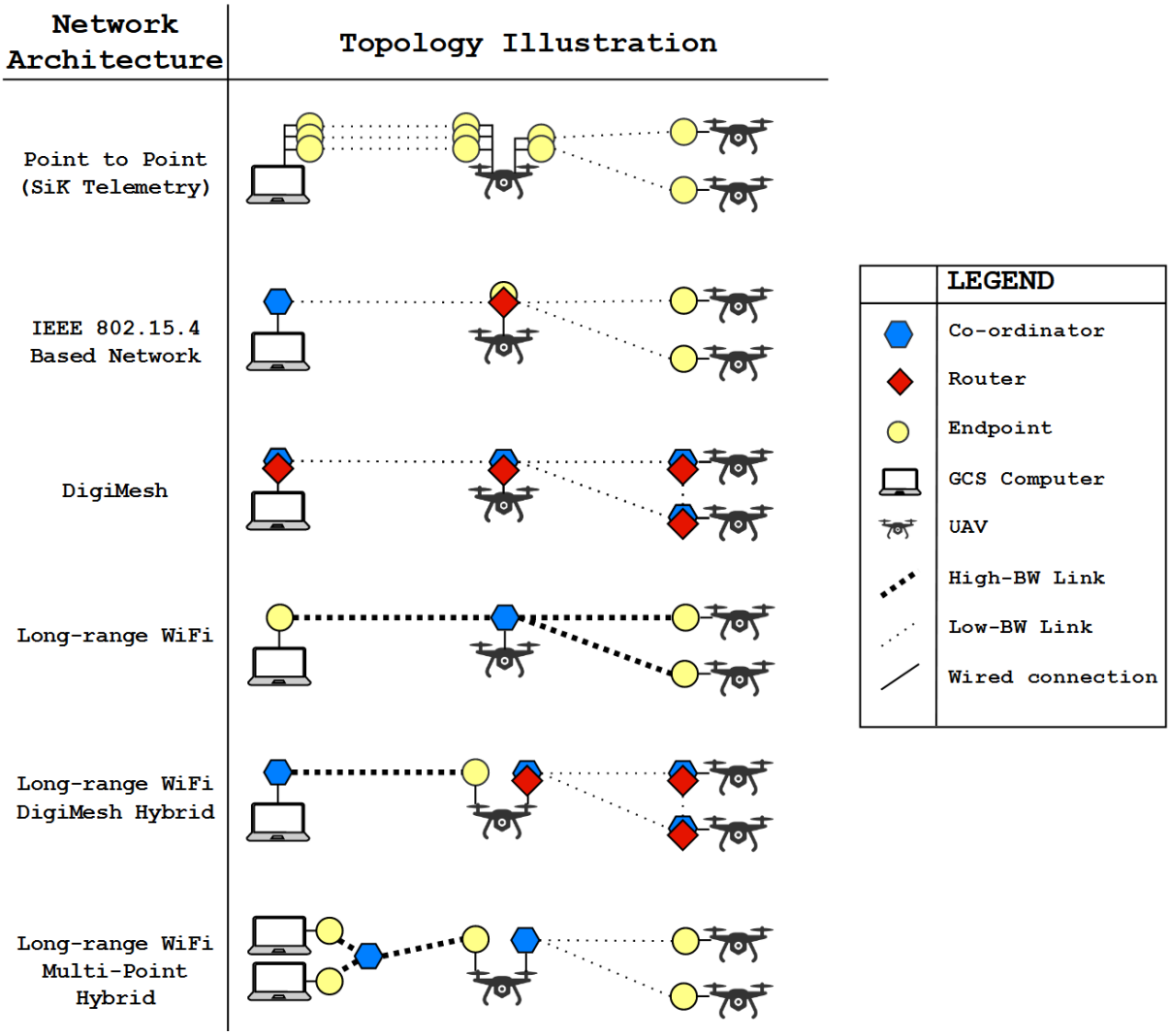


Figure 3.2: Visual comparison between different types of wireless networking architectures applied to an example BVLOS mission with one relay and two data collecting UAVs.

network. This is not a desirable implementation when looking forward to the project's stretch goals – specifically the ability to swap out relay UAV nodes during data collection missions. To elaborate, a relay UAV swap would be compromised from a reliability and safety perspective, because the GCS computer can only be wirelessly connected to one relay UAV at a time due to the network topology limitations. An alternative would be to use another GCS computer to connect to the newly launched relay UAV, although this comes with its own practical limitations as mission continuity would be broken with data split between two devices. Long-range WiFi could prove useful in BVLOS data collection missions, but given the limitations of one coordinator per network, it would work best in a hybrid network, not as a standalone solution.

The Wireless Research Centre (WRC) provided readily available equipment to pursue a network architecture consisting of DigiMesh using XBee S3B Pro radios, and long-range WiFi using Ubiquiti Rocket M2 transceivers. It was recognised that the purely DigiMesh network would be the simplest implementation for a proof-of-concept (PoC), and that work on this could be

easily adapted for a hybrid network with long-range WiFi connecting the GCS and relay UAV. Therefore, the prototype design for BVLOS UAV operations would implement a DigiMesh network using XBee S3B Pro radios for the communication of MAVLink telemetry data, with the initial design leaving room for various improvements and optimisations in future iterations.

3.2 Engineering Challenges & Modular Design

Proceeding with the decision to incorporate a DigiMesh network into the BVLOS communications prototype, the engineering challenges need to be identified for replacing SiK telemetry radios with XBee radios. Three different approaches were identified for the integration with the UAV, all with varying degrees of complexity associated: modifying PX4 firmware to support XBee radios with DigiMesh for BVLOS missions; designing a bespoke adapter to interface the XBee directly to the telemetry port; or connect both the Pixhawk and XBee to a companion computer that is running bespoke software to bridge the connection between Pixhawk and XBee. The latter seemed to be the clear choice when weighing up the complexity of the three options and development time. This is because it involves no hardware design, nor does it require the modification of professionally developed embedded software which would likely result in compatibility issues with future versions of the firmware. Using a companion computer and designing a middleware script follows modular design philosophy, whereby none of the pre-existing hardware or software intended for use requires modification.

QGroundControl, and other GCS software also does not support directly interfacing with XBee radios running DigiMesh firmware. Two different approaches were identified to enable this interface: modify a build of QGroundControl to support XBee radios, or write a middleware script to bridge the interface between the two modules. Continuing with the theme of modular design, another middleware script was developed to operate on the GCS computer, bridging the link between QGroundControl and the XBee radio. QGroundControl supports MAVLink connections through USB, transmission control protocol (TCP) and user datagram protocol (UDP) interfaces, expecting one UAV per interface. In the case of two software applications connecting to one-another locally on the same device, UDP is the best choice due to the smaller overhead when compared to TCP (8 vs 20 bytes) and UDP does not wait for acknowledgements, whereas TCP does [41, 42]. Although given the maximum DigiMesh network bandwidth of 92kbit/s, small differences in performance like that of TCP and UDP will be negligible on modern hardware. Therefore, the GCS middleware script will interact with multiple UAVs over a single XBee radio, and for each UAV a unique TCP or UDP port will be used to communicate MAVLink data between the script and QGroundControl.

3.2.1 Interfacing Modules

It is important to visually represent how key modules are interfaced with one-another, and where each module fits into the overarching system, in this case for both the GCS and relay/data collection UAV.

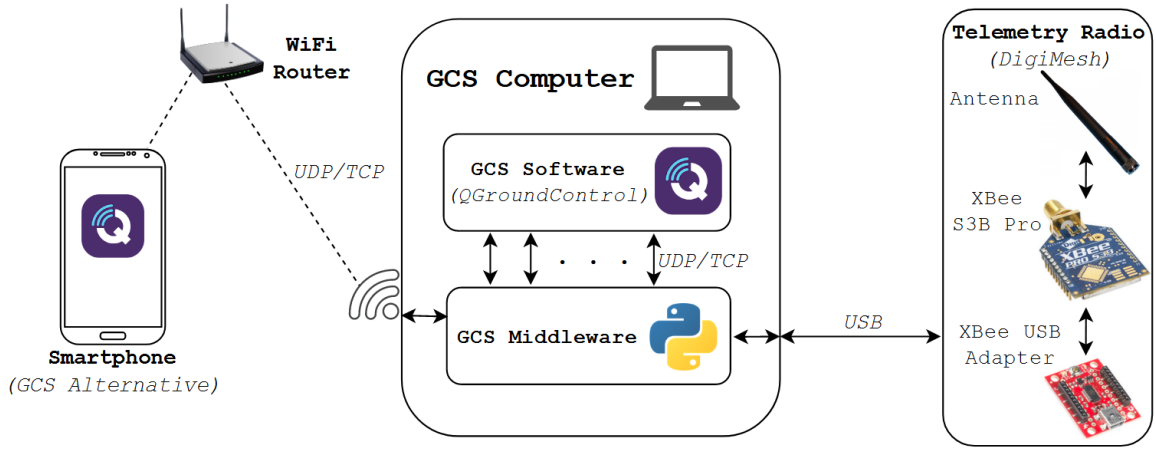


Figure 3.3: Modular overview of a ground control station using a XBee S3B Pro 900MHz radio, with a Python middleware script to correctly interface QGroundControl to the radio.

The GCS could feasibly be any modern computing device running an OS that supports the Qt GUI framework (Windows, Linux, macOS, iOS, Android), as QGroundControl is a cross-platform solution. Figure 3.3 illustrates that even a smartphone can feasibly be used as an alternative to a laptop or tablet, although most smartphones will not be capable of interfacing directly with an XBee radio. Therefore, for a smartphone to be used as a GCS, a TCP or UDP connection to a MAVLink device must be established on the same WiFi network as the primary GCS computer, which is connected to an XBee radio and running GCS middleware.

Figure 3.3 also illustrates how data is manipulated between QGroundControl and the XBee radio. Consider the same BVLOS mission scenario as seen in the topology diagrams of Figure 3.2 (one GCS, one relay UAV and two data collection UAVs). The XBee radio belonging to the GCS would be sending and receiving three MAVLink streams over the DigiMesh network, and the GCS middleware script will use three separate UDP sockets to communicate with QGroundControl.

The WRC and SERC's NAVI UAVs have a far more complicated modular overview than the GCS. Each of the NAVIs are a large quad-rotor UAV, fitted with a Pixhawk 1 flight controller, GPS module, Intel NUC5i7RYH, XBee S3B pro, power management unit (PMU), four brushless DC motors, and four electronic speed controllers (ESCs). For clarity of illustration, numerous other peripherals that are typically connected to the flight controller have been omitted from Figure 3.4, because they are not a critical component for BVLOS missions, they do not have an impact on the communication of MAVLink telemetry data. Examples include safety switch, buzzer, light emitting diodes (LEDs), battery monitor, redundant GPS, cameras, other

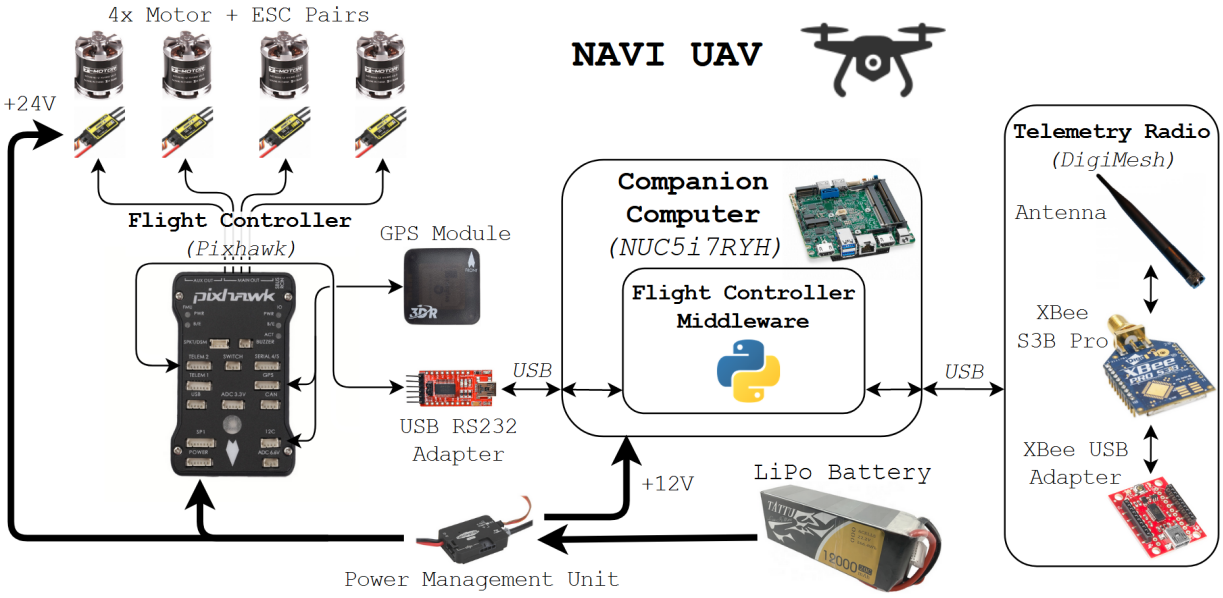


Figure 3.4: Modular overview of the NAVI UAV, with emphasis on the interface between the Pixhawk flight controller and XBee radio (via the Intel NUC5i7RYH companion computer).

transceivers, and other additional sensors. However, this illustration still provides a modular overview of each of the key electronic components on the NAVI.

Digi, the developers of the XBee radio and DigiMesh, also offer open source application programming interface (API) for their devices in a variety of popular programming languages, including C, Python and Java. ArduPilot has also released a Python MAVLink library, including a tool for generating a MAVLink API in a variety of other popular languages. Given that correct functionality, development simplicity, along with language preference, Python was the language chosen for both middleware scripts in the PoC design, despite the slight compromise in code performance. Therefore, middleware scripts will be written in Python, and two third-party libraries will be utilised to reduce complexity and accelerate development time: `pymavlink` [43] will be used to interface with both the Pixhawk and QGroundControl, and `digi-xbee` [44] for communicating with the XBee S3B Pro.

3.3 GCS middleware

Starting with the GCS middleware, the purpose of the script is to enable the use of the XBee S3B Pro radio using DigiMesh firmware. QGroundControl and other open-source alternatives only provide official support to radios that run SiK firmware for point-to-point telemetry links. While there is no official support for the XBee or DigiMesh, implementing the desired functionality in Python is a straight-forward task with the help of the `pymavlink` and `digi-xbee` libraries. Data from telemetry links will be encapsulated within DigiMesh packets, so data will be extracted from DigiMesh packets and routed to QGroundControl using a UDP port. The same process will also be carried out in the reverse direction, where data from QGroundControl

is assigned to the payload of DigiMesh packets and transmitted to the correct XBee radio.

In a telemetry link, the vast majority of data traffic is in one direction, from the UAV to GCS. QGroundControl (and alternative GCS software) transmits a MAVLink heartbeat at a rate of 1Hz, one to each UAV connected. All other traffic in this direction is the result of a manual command from the GCS, such as the sensor calibration process, uploading a mission map, and changing firmware parameters, etc. Therefore, the GCS middleware script does not have to alter data-rates to adhere to the DigiMesh and XBee bandwidth limitations, meaning no transformative actions need to be carried out on MAVLink data streams. The primary task of this script is routing the correct UDP socket with QGroundControl to the corresponding UAV's XBee radio.

Rather than immediately interfacing the XBee radio and QGroundControl, the first objective in the development process was to establish a telemetry link using a standard SiK radio, routing all information through Python middleware. This approach meant that the behaviour of the telemetry radio link could be observed, and the majority of the software could be written prior to switching the SiK radio with the XBee. Soon after commencing development, it became apparent that the script was not functioning as intended; the UAV was tilted to prompt a visual change on the pilot's dashboard, but this appeared to update only once per second. Investigating the issue revealed that UDP receiving function will block the process until data has been received. Given that the only regularly scheduled message type (heartbeat) is transmitted at a rate of 1 Hz, the process can be blocked for up to 1 second. To work around this a separate thread was created with the purpose of receiving data over UDP, with new data stored in a first in first out (FIFO) buffer `queue_out` shared with the main process. Another thread was implemented for UDP transmission, along with a FIFO queue `queue_in` shared with the main process, leaving the main process to handle serial communication with the radio.

Revising the script with a thread for each of UDP receiving, UDP transmission, and radio interfacing, resulted in seemingly identical functionality to when the SiK telemetry radio is interfaced directly to QGroundControl. At this point in development, the software interface to QGroundControl had been implemented and key characteristics of MAVLink telemetry streams had been observed. The next stage in development involved replacing the SiK telemetry radio with an XBee, requiring edits to the code within the radio interface thread, along with additional start-up code unique to the XBee and DigiMesh. The first design iteration of for middleware for both GCS and UAV assumed that the GCS 64bit address was defined in the UAV code, and the 64bit addresses of each UAV was defined in GCS code. During start-up, the GCS middleware would discover all UAVs on the DigiMesh network, open a UDP socket with QGroundControl, and create a UDP receiving thread for each UAV in the network. However, this was later changed such that neither the GCS nor UAV scripts relied on predetermined addresses. Instead GCS–UAV connections were negotiated through a request from the UAV

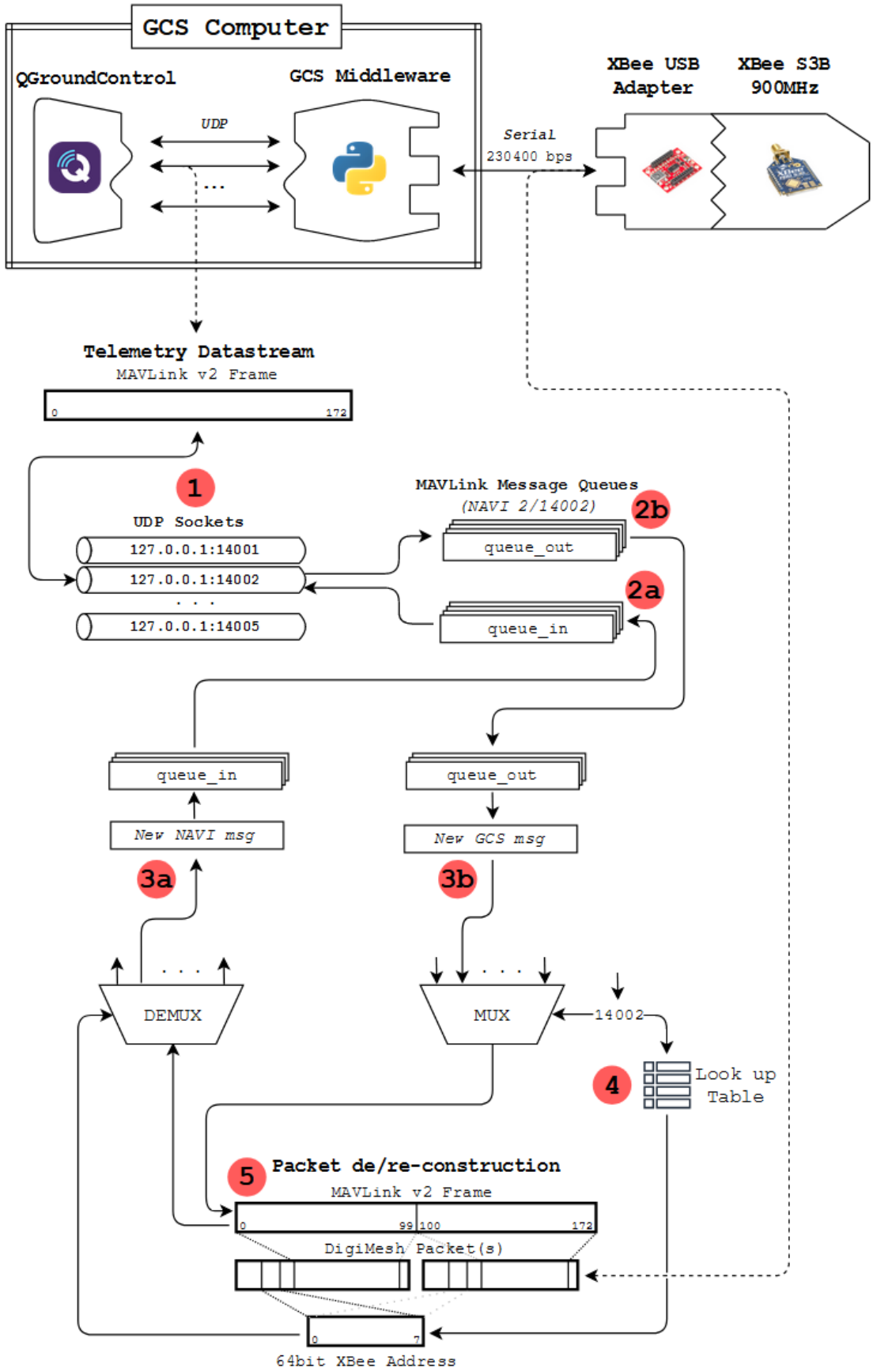


Figure 3.5: Diagram of hardware and software modules core to the GCS, including an illustration of the key processes taking place within the middleware script. Numbers inscribed in red circles indicate an important action taken by the middleware script.

script, and a response indicating whether the receiver XBee belongs a GCS or UAV. These changes allowed for UAVs to connect or disconnect from the GCS at any point throughout operation, with the script dynamically opening and closing corresponding UDP sockets and receiver threads. Therefore, $n + 2$ threads will be running within the GCS middleware, where n is the number of UAVs on the DigiMesh network. One thread dedicated to interfacing with the XBee, one thread for UDP transmissions, and n threads for UDP receiving.

The following list provides a description of actions or processes labelled in Figure 3.5.

1. A MAVLink packet 173 bytes in length (including overhead) is illustrated in Figure 3.5, v2 message signing has been disabled as securing the link from hijackers will be a low priority in Antarctica. MAVLink packets are transmitted between QGroundControl and the middleware script over UDP. The middleware script is running on the same hardware as QGroundControl, so the IP address for each UDP socket is local-host address `127.0.0.1`, and a unique port is assigned to each UAV on the DigiMesh network. Ports have been arbitrarily assigned using the following equation: $port = 14000 + n$, where n is the NAVI ID (1-5).

2. (a) While only a singular incoming queue (`queue_in`) is portrayed for this step in Figure 3.5, UDP transmissions to QGroundControl of all sockets are handled using one thread. This design decision was made because there is no blocking method or function in this process, and therefore the transmission of important telemetry data from UAVs will not undergo any meaningful delays. Messages from `queue_in` are transmitted in order from oldest to newest, as all queues in this script are treated as FIFO buffers.

   (b) Contrary to the transmission of MAVLink data over UDP, one thread is assigned to each individual UDP socket for receiving data. Receiving method (`socket.recv_msg()`) blocks the entire process until data has been received, which can take up to 1 second on a standard telemetry link. This is due to heartbeat MAVLink messages being the only scheduled message type (at a rate of 1 Hz) from QGroundControl to UAVs. However, when manual commands are being issued from QGroundControl, more data will be received from QGroundControl via the UDP socket accordingly. New messages from a UDP socket are appended to the back of the FIFO buffer `queue_out`.

3. (a) Newly received telemetry data (in the form of MAVLink frames) are encapsulated within XBee packets, the header of which indicates a 64bit address unique to the sender. This address is used to determine the correct destination FIFO buffer for new frames. This process is illustrated as a demultiplexer (DEMUX) in Figure 3.5, where the 64bit source address selects the correct output line for incoming frames, which leads to the corresponding `queue_in`.

   (b) In the reverse direction, a loop iterates over each `queue_out` to check whether there are MAVLink frames in the queue. If so, each frame is sent through to the portion of

code illustrated in steps 4 and 5. This process has been illustrated using a multiplexer, where each of the inputs correspond to an outgoing queue, and the UDP port number is used to select the corresponding queue to be emptied and prepared for transmission via XBee radio.

4. With the UDP port as a key, a look-up-table (LUT) is used to obtain the 64bit XBee address of the corresponding UAV. This is used in the header of DigiMesh packets to ensure packets are delivered to the correct XBee radio on the network.

5. For the transmission of MAVLink telemetry over a DigiMesh network, MAVLink frames are broken into chunks of up to 100 bytes and subsequently transmitted as the payload of DigiMesh packets. The illustration in Figure 3.6 shows a 173 byte MAVLink frame, which is broken into a 100 byte and 73 byte chunk, then transmitted as the payload of to sequential DigiMesh packets. The synchronous transmitting method `xbee.send_data(receiver, data)` blocks the process until an acknowledgement is received from the destination XBee. An error will be raised if either: an acknowledgement is not received after the maximum number of rebroadcasts, or another process attempts to use the XBee radio while busy.

3.4 Flight Controller Middleware

The largest challenge presented by this middleware script was a method of lowering the data-rate from the Pixhawk before transmitting data back to the GCS. The Pixhawk TELEM2 port is configured for a 921600 baud serial connection, transmitting telemetry data at an observed rate of over 100kbps. A standard telemetry radio link is typically less than 15kbps and the maximum theoretical bandwidth of the DigiMesh network according the the data-sheet is 92kbps [32]. This prototype BVLOS communications system must support a bare minimum of two UAVs (one relay and one data collector), meaning the DigiMesh network will not be able to meet the bandwidth requirements unless some kind of decimation takes place in software. Therefore, for this script to operate effectively and serve a useful purpose, one of the arbitrary design goals was to have fully user-defined control over the data-rate of each telemetry link. This enables the DigiMesh network to handle multiple UAV's with ease, and it would also prove useful when suggesting and abiding by and exact set of telemetry data requirements. Chapter 6 will touch upon this concept and its significance when suggesting a set of guidelines for policy changes around BVLOS UAV operations.

Starting with an earlier version of the GCS middleware script that utilised the SiK radio, the first task was to devise and implement a method of decimating data from the Pixhawk to XBee. Since a fully user-defined rate for each type of message is desirable, a LUT for message rates using MSG ID as the key was constructed and stored in a file, along with other attributes unique to the UAV. A new thread was also created to receive MAVLink frames from the Pixhawk, and check whether they are due for scheduled transmission. If the MSG ID

indicates that the MAVLink frame is due for transmission it is then appended to the outgoing transmission queue, and the scheduler LUT is updated using the following equation:

```
schedule_lut[MSG_ID] = time.time() + telem_periods[MSG_ID]
```

This script also needs to handle other MAVLink message types that are not regularly scheduled. For example, when a telemetry link is first established between a GCS and UAV, the GCS will make a one-off request for a list of UAV parameters. If the request frame(s) are received via the XBee and forwarded through to the Pixhawk, the PX4 firmware will identify the requested parameters and respond accordingly. The MSG ID of these MAVLink messages do not overlap with the regularly scheduled set of MAVLink message types, and all of these response packets should be transmitted back to the GCS. Therefore, if a MSG ID is not present in the message rates LUT, then it should have priority and placed immediately into the transmission queue. After implementing and testing this solution, a small blacklist was created for MAVLink MSG IDs to ignore, as these types of messages were not appropriate to send to the GCS.

After implementing code that enabled user-defined rates for each type of MAVLink message in a telemetry link, testing with QGroundControl suggested that packet loss was phenomenally high (over 80%). Investigating this issue revealed that packets were not truly being lost, instead QGroundControl was interpreting large packet loss by comparing the difference in the SEQ byte between consecutive MAVLink frames. Since SEQ is incremented after consecutive MAVLink frames are generated in a telemetry stream, QGroundControl (and other GCS alternatives) calculate packet loss by comparing the difference in SEQ bytes between consecutive messages (equation: `packets_lost += (SEQ`$_n$ `- SEQ`$_{n-1}$`)%255 - 1`). Therefore, the cause of this issue was the implementation of the user-defined data rate software, as it had not accounted for SEQ byte continuity. To fix this issue, the middleware script was given another task prior to transmission – replacing the SEQ byte in the header of each MAVLink frame in the transmission queue. Along with this, the CRC value had to consequently be recalculated and the old CRC field replaced, due to the change in frame data.

Before integrating XBee radios, the first stage middleware scripts for both the GCS and UAV still using SiK telemetry radios and handling MAVLink data. Once data handling and routing was implemented correctly, SiK telemetry radios were replaced with XBee S3B Pro radios. After this replacement it was observed that the RADIO_STATUS message was no longer present in the telemetry link, this being the only difference noticed. Upon further investigation it was found that RADIO_STATUS messages are generated within SiK firmware, and are transmitted as a reply to HEARTBEAT messages from the GCS, as indicated in the SiK firmware GitHub repository [45]. To completely replicate the function of the SiK telemetry radio, the flight controller middleware script must construct a RADIO_STATUS message based on information that can be obtained from the XBee radio. This type of MAVLink message presents the GCS with the following diagnostic data: RSSI, remote RSSI, free transmit buffer space, noise, re-

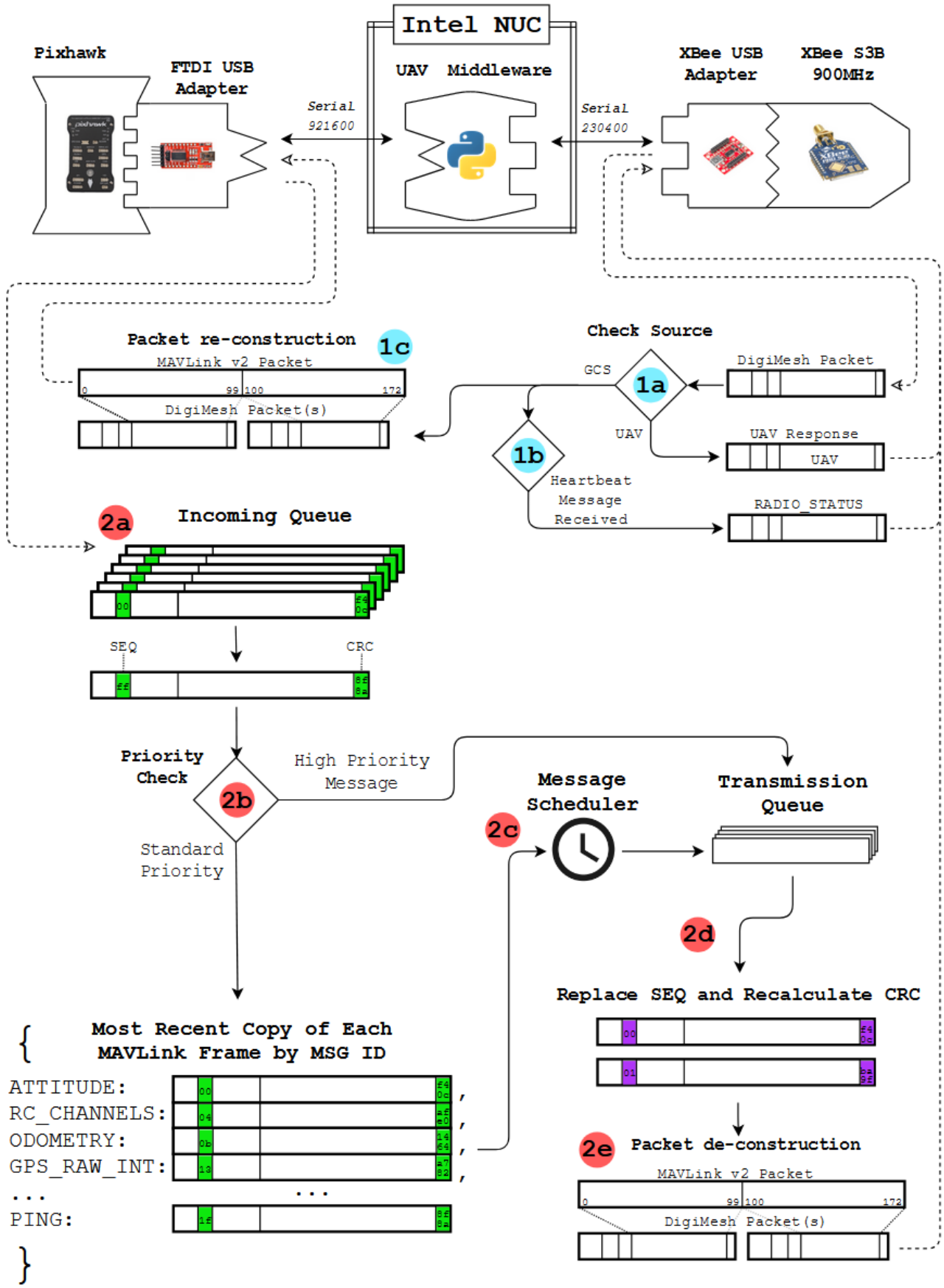


Figure 3.6: Adapter diagram illustrating each item of hardware involved in the communication of MAVLink data back to the GCS. The function of the middleware is also shown, where key steps are inscribed in circles indicate each of the step where MAVLink data is manipulated. Labels 2a – 2e indicate important actions carried out on MAVLink data sent from Pixhawk to XBee, and labels 1a – 1c indicate important actions in the opposite direction, from XBee to Pixhawk.

mote noise, number of received packet errors, and number of error corrected packets. The XBee S3B Pro cannot provide diagnostic information for all of the aforementioned data fields in a RADIO_STATUS message. Consequently, filler values are assigned to the following fields: noise, remote noise, and free transmit buffer space, while the the remaining fields will contain real diagnostic data [32].

The initialization code of the middleware script connects to the Pixhawk and XBee radio USB devices, then establishes a telemetry link with the GCS. As mentioned in Section 3.3, the flight controller middleware script initially had an address defined for the GCS XBee. This was later changed so that the flight controller middleware script would carry out DigiMesh network discovery, send a request to each XBee device in the network, and the response would indicate whether the XBee belonged to a GCS or UAV. After the initialization process is concluded, the middleware script advances to the MAVLink data handling code in order to replicate the SiK telemetry radio.

The important actions or processes labelled in Figure 3.6 are explained as follows.

1. (a) A new DigiMesh packet is read from the hardware buffer using the XBee Python API `xbee.read_data()` function, obtaining the first message from the XBee's FIFO buffer. The data and source address is checked to determine whether the transmitted message was from the GCS or another UAV. If the source is known to be the GCS then MAVLink data has been received and code progresses. Otherwise a response packet is transmitted to the sender, indicating that this XBee radio belongs to another UAV. Once the response message has been received, this XBee's 64bit address will be stored in a list of known UAVs and the other UAV will continue searching for the GCS XBee radio on the DigiMesh network.

   (b) If the message type is that of a MAVLink HEARTBEAT frame, the UAV must generate a RADIO_STATUS message to transmit back to the GCS. Regardless of whether the message is a HEARTBEAT or not, it will progress to step 1c.

   (c) Verification has been made that DigiMesh packet(s) have been received from the GCS, now the MAVLink packets are to be reconstructed. Since MAVLink v2 Frames can vary in size from 11 to 279 bytes, it can take between 1-3 DigiMesh packets to transmit an entire frame, as DigiMesh supports a maximum payload of 100 bytes [32, 20]. Once a MAVLink packet has been fully reconstructed, it is checked for errors then sent to the Pixhawk.

2. (a) A new message is read from the Pixhawk using the pymavlink `pixhawk.recv_msg()` function. Due to the high baud-rate of 921600 baud, a separate thread within the middleware script is dedicated to steps 1 and 2 to ensure that no MAVLink frames are lost due to a serial buffer overflow.

(b) Immediately after a message is received from the Pixhawk, the 3 byte message ID within the MAVLink v2 header is checked, determining the message type. This message type is then checked to see whether it should be ignored, prioritised or whether it is a regularly scheduled message type. No action is taken for ignored message types and priority messages are placed immediately into the outgoing transmission queue, advancing to step 2d. Otherwise, the code advances to 2c to handle message scheduling.

(c) If the MAVLink frame MSG ID is routinely transmitted in a telemetry link back to the GCS, then a dynamic LUT is checked to determine whether the MAVLink message type is due for transmission. If the message is due for a scheduled transmission, the MAVLink frame will be placed into the outgoing transmission queue, with the code advancing to step 2d. The time in the scheduler LUT will be updated based on a user defined time period for each type of MAVLink message (found in `uav_settings.json`).

(d) The sequence (SEQ) byte in the header of each MAVLink frame increments for every sequential frame, resetting to zero after the count hits 255. The purpose of this byte is to check how many MAVLink frames have been lost between two successfully received frames. Pseudo-decimation of the data-stream inevitably leaves gaps between the SEQ bytes of sequential messages in the outgoing queue, which if unaccounted for will result in a false error tally at the GCS. Therefore, the SEQ value is replaced to maintain a realistic error count on the GCS for accurately indicating the link quality, and consequently the last two CRC bytes have to be recalculated. The new values for SEQ and CRC fields are indicated with a colour change from green to purple.

(e) Altered MAVLink frames are now broken into sequential chunks up to 100 bytes long in order to fit into the payload of a DigiMesh packet. DigiMesh packet(s) of the chunked MAVLink frames are formed and sent to the XBee radio for transmission to the GCS using `xbee.send_data(remote, data)`, where `remote` is the object representing the GCS XBee radio, and `data` is the payload of up to 100 bytes.

3.5 Testing & Results

With the wireless communications design outlined and practically implemented for BVLOS UAV operations, the final stage involved testing and evaluating performance. This consists of quantifying throughput limitations of the DigiMesh network, error rates and packet loss, while range will be discussed in Chapter 4. The XBee Configuration and Testing Utility (XCTU) application, used to configure XBee devices, features frequently in this section due to it's network discovery GUI and throughput testing tool.

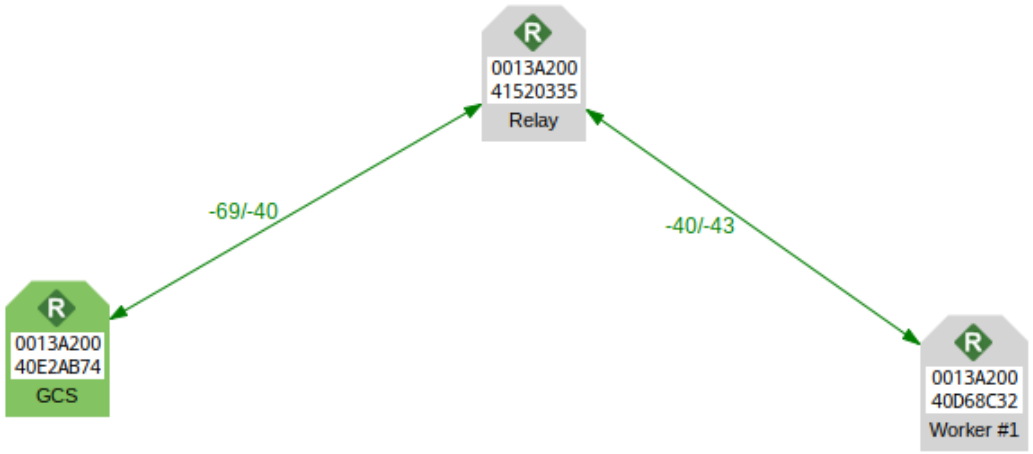
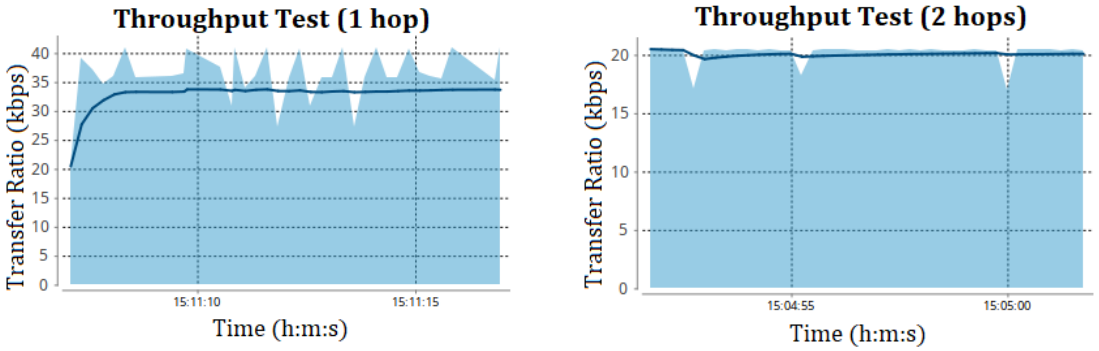


Figure 3.7: DigiMesh network configuration after carrying out a discovery process, generated using XCTU.

Figure 3.7 shows a DigiMesh network after the network discovery process, consisting of three XBee radios. Two of the XBee's (*GCS* and *Worker #1*) do not have a direct (1-hop) route of communication, and therefore must communicate via the third XBee (*Relay*), requiring an additional hop. When the *Relay* XBee receives a message from *GCS* that is explicitly addressed to *Worker #1*, the same message will then be re-transmitted by the *Relay* XBee. With this configuration, throughput tests were carried out over 1-hop and 2-hop bidirectional links.



(a) Single hop between GCS and Relay.

(b) Two hops between GCS and Worker (via Relay).

Figure 3.8: Data throughput graphs generated within the XCTU application's speed-test tool, where the light-blue indicates the instantaneous data-rate and the dark blue line indicates the average data-rate over time.

The bidirectional throughput average was reported as 33.64 kbps and 20.86 kbps for the 1-hop and 2-hop configurations respectively. A time-series plot of these results can be seen in Figure 3.8. It can be observed in Figure 3.8a that the 1-hop test yields an instantaneous throughput

of up to 41 kbps, around double the average reported for the 2-hop test shown in Figure 3.8b. Repeating the 1-hop test multiple times reported the same maximum value of 41 kbps, with the average also remaining at a similar value. These values for the 1-hop result pail in comparison data-sheet specification of 91.0 kbps for synchronous 1-hop transmission [32]. This disparity could possibly be as a result of both XBee's being configured as standard routers. Another possibility is that the channel mask limiting the available FHSS channels. The NZ general user radio license allows for used on 915 MHz – 928 MHz, whereas the XBee is 902 MHz – 928 MHz capable, and the full range of channels were used to obtain a 1-hop throughput of 91.0 kbps. Despite the speculation and changing these settings, repeated tests never yielded an instantaneous throughput greater than 41 kbps.

The PoC design that each data collection UAV can only communicate over a 2-hop connection with the GCS via the Relay, and all UAVs (including the Relay) are assigned the same telemetry rate. Therefore, the telemetry rate of each UAV can be calculated as

$$D_{TELEM}(n) = \frac{D_{DM}}{2n+1}, \tag{3.1}$$

where D_{DM} is the DigiMesh network throughput, and n is the number of data collection UAVs. If D_{DM} is set to the maximum of 41 kbps, D_{TELEM} is calculated to be 13.67 kbps for $n = 1$, 8200 bps for $n = 2$, and 5857 bps for $n = 3$.

3.5.1 Pre-Flight Considerations & Field Trials

Upon establishing a telemetry link between GCS and UAV, PARAM_REQUEST_LIST is sent to the UAV, which responds with 698 individual PARAM_VALUE messages. With the payload of each PARAM_VALUE message containing 25 bytes and the MAVLink v2 frame overhead of 12 bytes, this results in 25826 bytes or 206608 bits of data being transmitted from UAV to GCS upon start-up. As the number of data-collector UAVs increases this process takes an exceedingly long time when compared to the standard SiK PtP link, which would often load the parameter list in 1-2s. A practical solution was to initiate telemetry links sequentially over 1-hop communications, and assigning priority to all PARAM_VALUE messages. However, with these measures in place the pre-flight process remained longer and less reliable when compared to a PtP telemetry link standard using SiK radios.

Proceeding past the parameter request process that occurs whenever a telemetry link is established, the link is now in a state where regularly scheduled messages are exchanged between GCS and UAVs. If RC message types are ignored due to manual flight not being feasible or practical in BVLOS missions, the data-rate for regularly scheduled MAVLink messages on the telemetry link drops to 11.10 kbps. This value was calculated by identifying each regularly transmitted MAVLink message type, observing the average transmission period for each message type, then using the period to divide the number of bits per MAVLink frame.

Regular Message Types of a Telemetry Link					
MAVLink ID	**T (s)**	**bits**	**MAVLink ID**	**T (s)**	**bits**
PING	10	208	COMMAND_LONG	1	360
VFR_HUD	0.625	256	HOME_POSITION	5	512
ATTITUDE	0.125	320	BATTERY_STATUS	2.5	384
ODOMETRY	0.825	1920	ATTITUDE_TARGET	1.25	392
TIMESYNC	1	224	SERVO_OUTPUT_RAW	2.5	264
ALTITUDE	2.5	352	ESTIMATOR_STATUS	5	432
HEARTBEAT[1]	1	168	EXTENDED_SYS_STATE	2.5	112
VIBRATION	5	352	LOCAL_POSITION_NED	2.5	320
SYS_STATUS	2.5	344	GLOBAL_POSITION_INT	0.5	320
RC_CHANNELS[2]	0.1	432	ATTITUDE_QUATERNION	1	352
GPS_RAW_INT	0.25	336	RC_CHANNELS_OVERRIDE[2]	0.1	240
SYSTEM_TIME	1	192	ACTUATOR_CONTROL_TARGET	1	424
HIGHRES_IMU	1.67	592	POSITION_TARGET_GLOBAL_INT	5	504

Table 3.1: Summary of a standard telemetry link between a UAV and GCS. Time periods were observed, and message sizes were obtained from MAVLink documentation [18].

Table 3.1 provides a summary of all regularly scheduled message types in a standard telemetry link. It should be noted that the bit values assume: message signing is not enabled (otherwise an additional 104 bits should be added), two CRC bytes are present, and no truncation of any trailing zeros (a feature of MAVLink v2). Since truncation is not factored in, the telemetry data-rate predictions calculated using Table 3.1 will be an upper-limit. Therefore, while these calculations will serve as a guideline, the telemetry rate will in practise be lower, due to the removal of trailing null bytes from payloads.

Data-rate monitoring was also added into both Python adapter scripts, reporting similar measurements to that of the XCTU throughput test. This was added to ensure that telemetry rates matched what was specified in `uav_settings.json`, to monitor packet loss and error rates, and finally the DigiMesh network throughput. Since XCTU is closed source, this additional monitoring confirms whether or not the settings and results of the throughput test align

[1]The GCS also sends a HEARTBEAT to each UAV ($T = 1$s), a RADIO_STATUS message is sent in response.
[2]Arbitrarily set at 0.1s as data is non-critical. Only present when RC is connected during field tests.

with what is reported in the middleware scripts when MAVLink telemetry data is being transmitted during a mission. One Relay and one data collector UAV ($n = 1$) connected to the GCS was used to verify the functionality of the middleware monitoring code. The defined telemetry rate was increased to the maximum theoretical value of the given configuration: $D_{DM} = 41$ kbps, $D_{TELEM} = 13.67$ kbps. Average network throughput appeared to stagnate at 35 kbps, slightly above the average reported during the XCTU 1-hop throughput test. This verified that synchronous transmission was used in the XCTU test, and that DigiMesh packet overhead was not factored into calculations In addition, this confirmed a negligible disparity in throughput whether variable or constant payload length is used. However, when the defined telemetry rate exceeded the practical data limitations of the network, transmit buffers began to overflow, resulting in a gradually increasing feedback lag and packet loss. Therefore, D_{DM} should be set to a value smaller than the maximum network capacity.

Figure 3.9: Mission map designed in QGroundControl, prior to uploading to a UAV for flight. Testing carried out over Ilam Fields, adjacent to University of Canterbury campus.

At this stage, lab testing with disarmed UAVs had concluded for the PoC system, with field trials being the next stage in testing. A simple $n = 1$ configuration was used (one relay and one data collector), where the relay was to remain either idle on the ground or maintaining a constant position in space. To ensure for safe and reliable flight, manual test flights were carried out using an RC for control while the telemetry link integrity, and middleware software was monitored. Arbitrarily setting $D_{DM} = 25$ kbps eliminated any accumulating lag, and the reported packet loss statistics (found in QGroundControl, based off of MAVLink SEQ byte) were comparable to a PtP telemetry link using SiK radios. This results in an D_{TELEM} value of 8333 bps for $n = 1$, with the message transmission periods adjusted accordingly such that this new telemetry constraint was met. It is also worth noting that lowering D_{DM} below the maximum network capacity is required to maintain the integrity of telemetry links, by accom-

modating for irregular/one-off exchanges between the UAV and GCS. For example, whenever a new way-point (or map of way-points) are sent to the UAV, there is a one-off exchange of information, effectively increasing the data throughput of the telemetry link until the exchange has finished.

Once the correct functionality of the middleware and telemetry links had been observed during manual flight, the next and final stage in the PoC was to upload and execute a pre-planned mission. A basic survey mission was created in QGroundControl, shown in Figure 3.9, and uploaded to the UAV after initiating the telemetry link. The only notable issue was the previously mentioned parameter request sequence, which carried out upon establishing a telemetry link between GCS and UAV. Otherwise, packet loss over DigiMesh was observed to be similar to that of a PtP link with SiK radios, and there appeared to be no reduced functionality or feedback. The only exception being that the XBee radio hardware used is not capable of determining: noise levels, number of errors corrected, or free space remaining in the transmit buffer. This means that RADIO_STATUS frames use placeholder values for: transmit buffer space, local noise level, remote noise level, and number of errors corrected. While the use of placeholder values is not ideal, the RADIO_STATUS message still includes useful values for: local RSSI, remote RSSI, and number of errors received, all of which are obtained from the radio hardware.

3.6 Discussion

Throughout the process of designing and testing the PoC BVLOS communications architecture for UAVs, a variety knowledge and practical experience was attained. This will help further future iteration of the wireless communications system design. Many practical issues were encountered and fixed accordingly, eventually resulting in the following milestones being achieved during the development of this first prototype BVLOS system:

- Software enabling the use of DigiMesh via XBee radios for UAV telemetry connections, a replacement to the point-to-point topology offered by SiK radio.
- Two UAV's were successfully flown simultaneously over a DigiMesh network, both receiving commands from one GCS.
- MAVLink telemetry connections between GCS and UAV via a relay radio.
- User-defined rates for each type of MAVLink message in a standard telemetry link.

Despite DigiMesh's theoretical maximum theoretical throughput of 91.0 kbps [32], this value could not be achieved through any configuration of parameters, even with just two radios in the DigiMesh network. The maximum practical throughput was found to be 33-35 kbps, observed through the XCTU application and verified through diagnostics added to both Python

middleware scripts. This limitation first became apparent when multiple GCS–UAV telemetry connections were established simultaneously, as parameter requests from the GCS result in a large number of priority messages from each UAV, temporarily exceeding the intended data-rate of the telemetry connection. Although a method was implemented for data-rate decimation of telemetry link, the network constraints eliminated the possibility of transmitting additional sensor to the GCS during a mission. For example, snow thickness data would have to be stored on board, and no images, let alone video, could be transmitted to the GCS. Data-rate constraints were further exemplified when communications between GCS and UAVs were indirect via a relay XBee, with bottle-necking occurring between GCS and relay. This effect on data-rate was described in Equation 3.1, where in the simplest case of $n = 1$ and $D_{DM} = 25$ kbps, the telemetry rate limit for each UAV was calculated to be $D_{TELEM} = 8333$ bps.

The standard PtP network architecture for telemetry links had been replaced with DigiMesh, preserving functionality at the expense of telemetry link throughput. Therefore, the first objective derived from the aim of this thesis had been accomplished – a PoC system design and mesh network implementation to enable BVLOS flight via a relay UAV. Following iteration of the BVLOS communications system would involve modifications to the network architecture and replace the radio hardware associated. The highest priority would be placed on increasing the bandwidth of connection between GCS and relay UAV to eliminate network bottle-necking. Establishing maximum operating ranges, optimising the Relay – Data Collector network, and definitions for minimum telemetry rates will also be investigated in following chapters.

Chapter 4

Physical Layer & Spatial Considerations

With a PoC design for BVLOS data collection missions, the focus now shifts towards identifying characteristics of the wireless channel, along with spatial optimisations and constraints. This chapter will cover all of the considerations made for BVLOS data collection missions in Antarctica, demystifying any environment specific unknowns prior to the final design iteration. Unique aspects of the environment and their implications on wireless communications will be investigated, such as humidity, temperature, and electromagnetic reflections from the snow/ice interface. The two-ray channel model will be applied for over-ice BVLOS communications between nodes, and a data collection method will be investigated in an attempt to verify the validity of the parameters of this channel model, specifically the ice reflection coefficient. Finally, range estimations will be made for both the PoC system that utilises XBee S3B radios, and the next iteration of the design involving Ubiquiti Rocket M2's.

4.1 Modelling the Ross Sea Wireless Channel

Data collection missions are anticipated to take place over the Ross Sea, with the operational GCS being situated upon an icebreaker, known as the S.A. Agulhas II. Assuming free space path loss (FSPL), the maximum distance between transceivers can be determined by setting P_{RX} to the receiver sensitivity P_{RS}, factoring in noise components then rearranging for d. However, there are many other factors to consider when estimating the maximum practical distance for reliable wireless communications. Multi-path propagation effects, Fresnel zone obstructions, and hardware condition to list a few, can all negatively impact the received signal, consequently lowering maximum range. Research also suggests that environmental characteristics, such as temperature and humidity can have an effect on wireless communications. Some of these effects can be modelled and approximated. Other effects can merely be speculated as a source of loss without practical trials. This section will assume that the snow covered sea-ice is the only significant reflective surface during data collection operations. Therefore, the two-ray channel model will be used to approximate propagation from the wireless channel.

4.1.1 Physical & Spatial Characteristics

The following is a list of assumptions made to simplify the simulation and approximation the channel model:

- Snow covered sea-ice is the only significant reflective surface.
- Snow/ice permittivity remains constant, despite the variation in snow depth and ice thickness.
- The snow/ice surface is flat and perfectly level with the sea.
- Omnidirectional antennas have a perfectly spherical radiation.
- Signals are negated outside of the vertical beamwidth of a directional antenna.
- Signal gain within the beam of a directional antenna is uniform.

Wireless Channel Parameters		
Name	**Symbol**	**Value**
GCS Altitude	h_g	15 m
Data Collector Altitude	h_d	5 m
Snow/Ice Permittivity	ϵ_r	3.17 [16]
Antenna Gain	G_{TX}, G_{RX}	2.1 dBi
Transmit Power	P_{TX}	24 dBm
Receiver Sensitivity	P_{RS}	-101 dBm
Hardware Losses	L_{TX}, L_{RX}	0 dB
Miscellaneous Losses	L_m	0 dB
Carrier Frequency	f_c	915 MHz
Horizontal Distance	d	*Variable*
Relay Altitude	h_r	*Variable*

Table 4.1: Constants assumed for approximating the Ross Sea wireless channel model when using XBee 900 MHz radios.

The values for constraints and assumptions are summarised in Table 4.1, providing all of the constant values required to determine FSPL and a two-ray channel model. It is assumed that the snow-ice surface is smooth, flat, maintains no tilt, and has a relative permittivity value of

$\epsilon_r = 3.17$. Therefore, r, r', l, $\Delta\phi$ and ϕ_i can all be determined when provided values of the spatial parameters h_r, h_d, and d. These values are consequently used to calculate Z, which is required to calculate the reflection coefficient R, and finally the received power P_{RX} of the two-ray model. As indicated in Table 4.1, P_{RX} is effectively a function of d and h_r, with other parameters either being constant or a function of the aforementioned constants and/or variables. Relay height and horizontal distance are varied in simulations, such that: 10 m $\leq h_r \leq$ 1000 m, and 10 m $\leq d \leq$ 100 km. Logarithmic spacing will be used for generating graphs. Points of intersection are found initially using logarithmically spaced arrays, then a small 1m resolution array will be generated for a more accurate estimate.

4.1.2 GCS-Relay Channel Model – Constrained FSPL

Since the GCS is assumed to be stationary and not constrained in terms of either energy supply or equipment size, it is feasible to use a highly directional antenna. This simple change would improve link performance in two important ways. Most obviously, the gain of transmitted and received signals at the GCS would be increased by a significant factor. Secondly, the directionality can also significantly mitigate the multi-path effect of the two-ray channel model. This is the case if the following condition is met:

$$\phi_i \geq \frac{\phi_{bw,v}}{2} - \phi_{az} \tag{4.1}$$

where ϕ_i is the incident angle, $\phi_{bw,v}$ is the vertical beamwidth, and ϕ_{az} is the angle of azimuth. For example, if a yagi directional antenna with $\phi_{bw,v} = 50°$, pointed directly at the relay UAV such that $\phi_e = \phi_{az} = 20°$, the signal strength of the reflected signal will be significantly diminished if $\phi_i \geq 5°$.

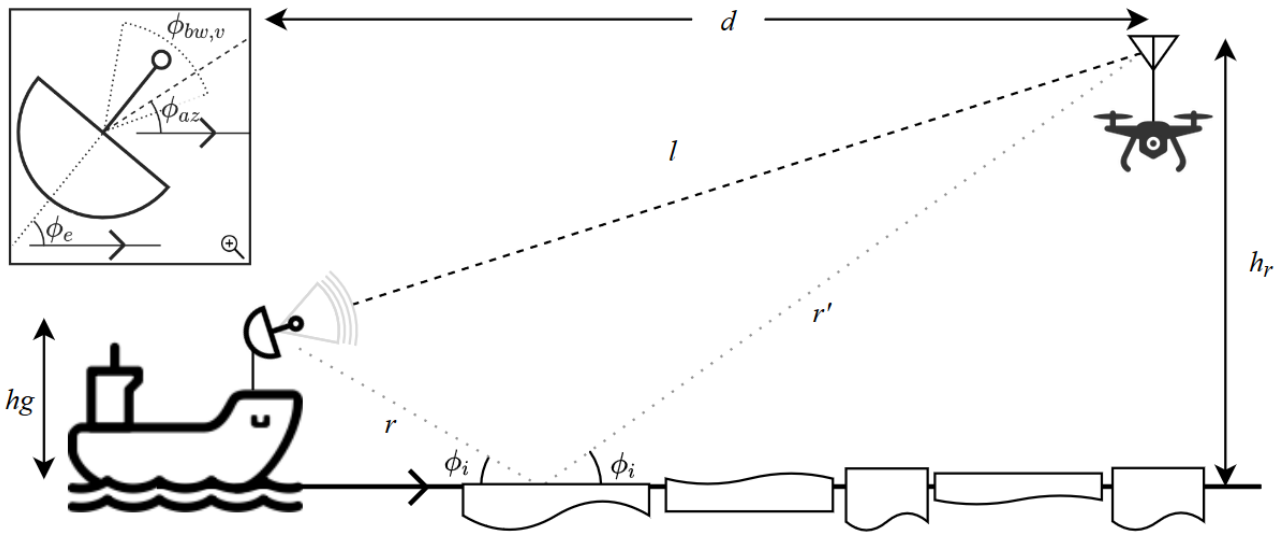


Figure 4.1: Spatial diagram of the GCS – Relay UAV configuration, along with a close-up of a directional antenna, both annotated with relevant parameter symbols.

This concept is illustrated in Figure 4.1 with key parameters annotated. If perfect antenna tracking is assumed based on the relay UAV's position, ($\phi_e = \phi_{az}$), then the two-ray model's

interference will still apply when the condition described in Equation 4.1 holds. Since both ϕ_i and ϕ_{az} are a function of the UAV spatial position, an equation can be derived to describe when the two-ray model is present with antenna tracking at the GCS. However, when $\phi_{az} \geq \phi_{bw,v}/2$, then there is no such ϕ_i for which the reflected signal is not mitigated. This is implied in Equation 4.1 and Figure 4.1, as ϕ_i must be a positive angle. An alternative configuration to ensure mitigation of the multi-path effects of the two-ray model, is to set the elevation angle such that $\phi_e = \phi_{bw,v}/2$. Applying this method is cheaper and simpler than using an antenna tracker, although it introduces the following spatial constraint $0 \leq \phi_{az} \leq \phi_{bw,v}$.

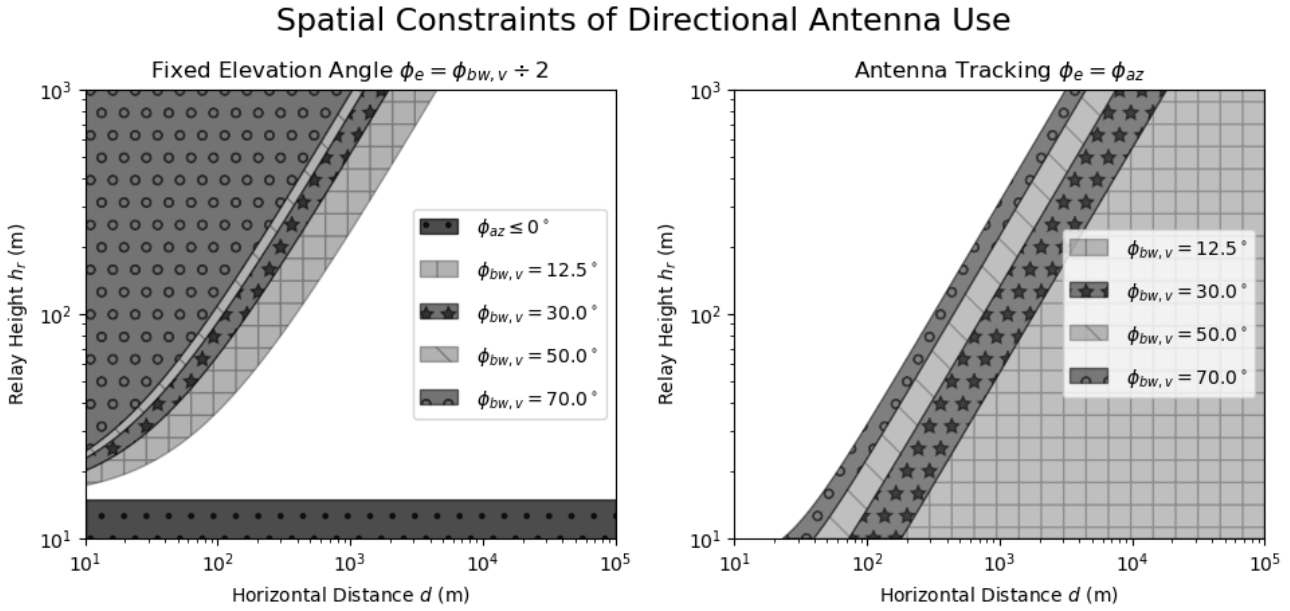


Figure 4.2: Simulation of spatial constraints for relay flight subject to the GCS – relay UAV communications link. For fixed elevation angle, shaded regions indicate spatial positions outside of the directional antenna's beam. For antenna tracking, shaded regions indicate where two-ray model interference applies.

The spatial constraints/considerations of switching from an omnidirectional antenna to a directional antenna are illustrated in Figure 4.2. Four separate cases for vertical beamwidth, of $\phi_{bw,v} = \{12.5°, 30°, 50°, 70°\}$ have been considered in the simulation. As expected, fewer spatial constraints apply for fixed angle elevation when using a directional antenna with a larger $\phi_{bw,v}$. The opposite is true for antenna tracking, where fewer spatial constraints apply with a smaller $\phi_{bw,v}$.

As mentioned in the discussion of the PoC system, the GCS – relay link requires higher throughput capabilities. Therefore, readily available 2.4 GHz Ubiquiti Rocket M2 (URM2) radios will be used for the link, with no changes being made to the inter-UAV network configuration. Assuming the GCS – relay link is not subject to multi-path propagation effects, FSPL can be used to theoretically determine a maximum operational distance.

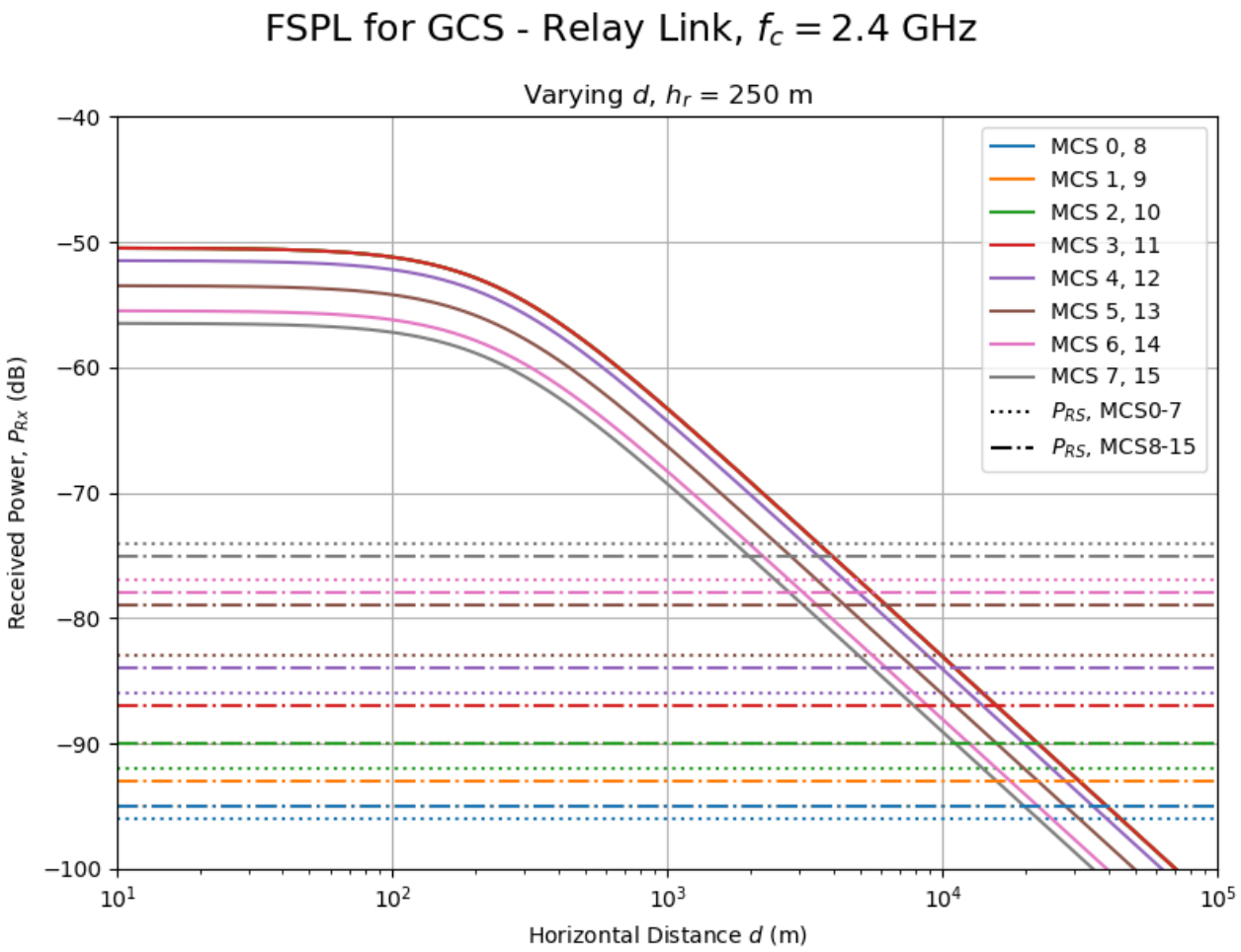


Figure 4.3: Simulation of FSPL with URM2 2.4 GHz transceivers for the supported MCS (0-15).

MCS Maximum Distance			
MCS	$d_{\max}$	**MCS**	$d_{\max}$
0	44.40 km	8	39.57 km
1	39.57 km	9	31.43 km
2	28.02 km	10	22.25 km
3	22.25 km	11	15.75 km
4	12.51 km	12	9.94 km
5	7.03 km	13	4.43 km
6	2.79 km	14	3.13 km
7	1.75 km	15	1.97 km

Table 4.2: Values for which P_{RX} intersects the P_{RS} threshold, indicating maximum theoretical distance d for each of MCS 0-15 based on hardware specifications of the URM2 [46].

With $f_c = 2.4$ GHz, a directional antenna $G_{TX} = 6$ dBi, and a fixed $h_r = 250$ m, Figure 4.3 depicts P_{RX} as d increases. Since the URM2 supports MCS 0-15, each of which having a different combination of values for P_{TX} and P_{RS}, these were plotted to illustrate the trade-off between MCS selection and range. The P_{TX} values of MCS 0-7 align with that of MCS 8-15, although the same cannot be said about P_{RS}. MCS 0-3, 8-11 all share a P_{TX} of 28 dBm, and hence are superimposed on-top of each-other. P_{RS} for MCS 1 and 8 share a value of -95 dBm, and both MCS 3 and 10 share a value of -90 dBm. All other MCS have a unique P_{RS} and are therefore visibly distinguishable.

Reiterating on the concept covered in Table 2.3 - as both the coding rate CR and bits per symbol N_{BPSC} increase, EVM decreases. This is the biggest attributing factor that results in P_{RS} increasing with R and N_{BPSC} as MCS varies. Therefore, the maximum range for wireless communications diminishes with EVM, as confirmed in Table 4.2.

MCS Maximum Distance, Factoring in Receiver Tolerance

MCS	$d_{\max}$		MCS	$d_{\max}$	
	$P_{RS} + 2$dB	$P_{RS} - 2$dB		$P_{RS} + 2$dB	$P_{RS} - 2$dB
0	35.27 km	55.90 km	8	31.43 km	49.82 km
1	31.43 km	49.82 km	9	24.97 km	39.57 km
2	22.25 km	35.27 km	10	17.68 km	28.02 km
3	17.68 km	28.02 km	11	12.51 km	19.83 km
4	9.94 km	15.75 km	12	7.89 km	12.51 km
5	5.58 km	8.86 km	13	3.52 km	5.58 km
6	2.21 km	3.52 km	14	2.49 km	3.95 km
7	1.39 km	2.21 km	15	1.56 km	2.49 km

Table 4.3: Vales for which P_{RX} intersects the P_{RS} threshold, this time factoring in the ± 2 dB receiver tolerance.

One final consideration is factoring in the ± 2 dB variation receiver tolerance of the URM2 [46]. This seemingly small variation can significantly alter the maximum operational distance, as seen in Table 4.3. Using MCS 0 as an example, the theoretical value for $d_{\max}$ varies by up to 20 km in the aforementioned simulation conditions. The specified tolerance is likely random, or depends on factors that cannot be modelled without further hardware analysis. Therefore, a conservative assumption will be made: that the lower theoretical value will be the upper range limit for the link between GCS and relay UAV.

4.1.3 Inter-UAV Channel Model – Two-Ray Model

Directional antenna use is physically viable at the GCS, and can mitigate the effects of snow/ice reflections. However, equipping UAVs with directional antennas would impose significant physical limitations that would hinder flight duration and aerodynamic performance. Instead omnidirectional antennas will be equipped to both the relay and data collection UAVs, for 915 MHz inter-UAV communications. Assuming a perfectly uniform radiation pattern, and the previously mentioned simulation assumptions, this implies that inter-UAV communications are subject to the two-ray channel model.

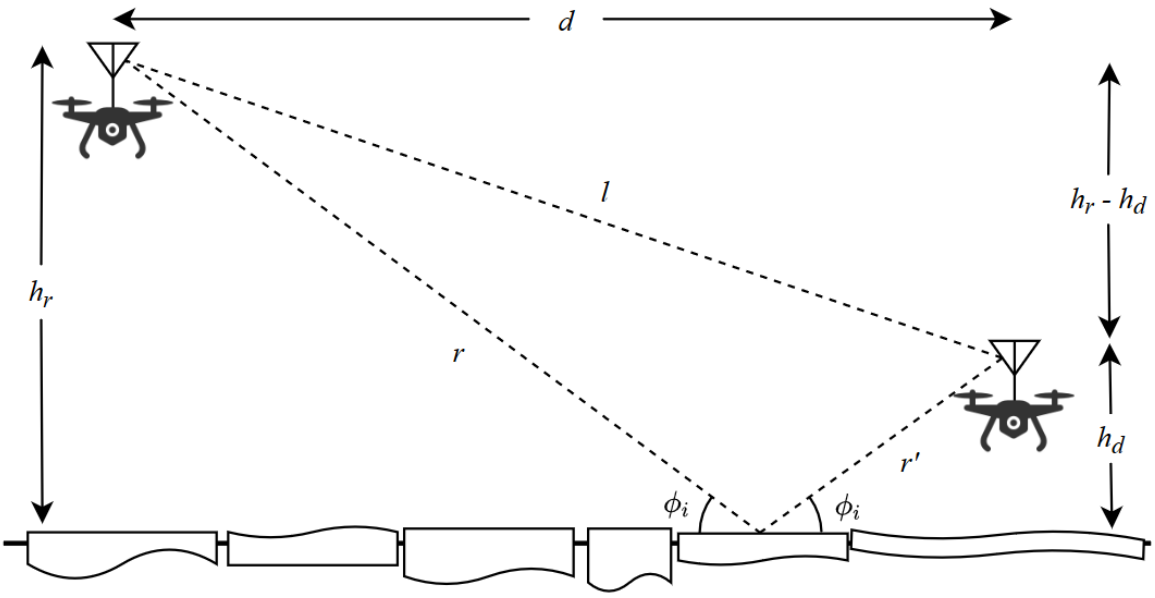


Figure 4.4: Spatial depiction of the two-ray channel model, annotated with key parameters.

For relay – collector inter-UAV communications, all antennas are vertically polarised and a uniform emission radiation pattern for primary and secondary signal paths is assumed for calculating the Z value in Equation 2.4. Figure 4.4 illustrates the two-ray model, specifically applied to inter-UAV wireless communications. The parameters shown in Table 4.1 are again used for this simulation.

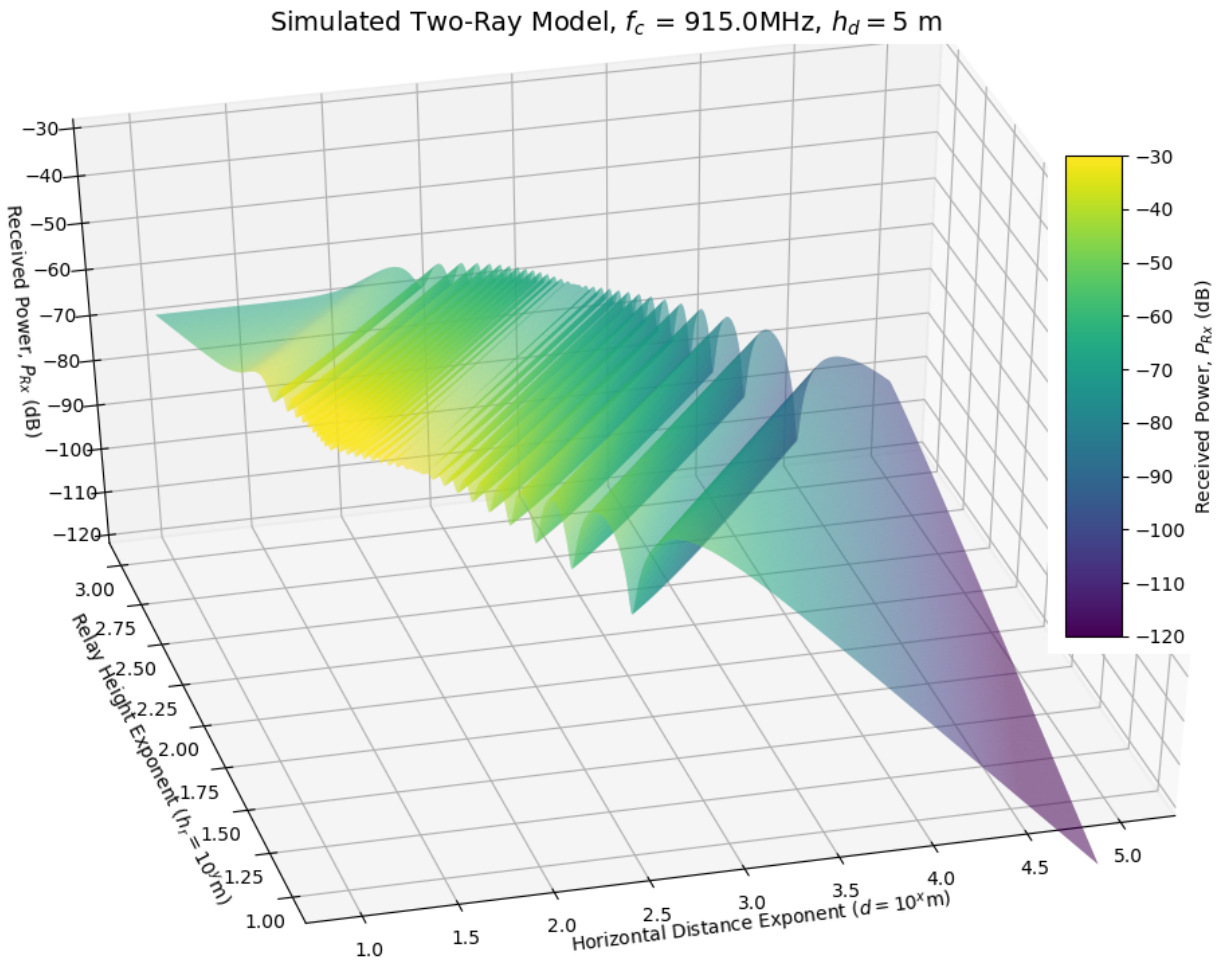


Figure 4.5: 3D visualisation of the two-ray model simulation, varying both d and h_r on the x-axis and y-axis respectively.

Figure 4.5 is a simulation of the two-ray model, generated using the constants specified in Table 4.1 while varying horizontal distance, d, and relay altitude, h_r. A sinusoidal effect can be seen on the surface diagram, where peaks indicate maximum constructive interference, and troughs indicate maximum destructive interference. If h_r is set to 250 m, it can be observed that the magnitude of the interference effect is minimal around $d = 450$ m. As d increases past this point, the frequency of the sinusoidal effect decreases and the magnitude of interference increases to a much more significant point. For example, the third to last local minima occurring at $d = 2530$ m, has a lower P_{RX} than the final local maxima occurring at $d = 11890$ m, with P_{RX} of -80.6 dB and -80.2 dB respectively. Increasing the horizontal distance by 470 m to where the third to last local maxima occurs $d = 3000$ m, provides a +12.1 dB improvement in P_{RX}, to -68.5 dB.

The aforementioned observations are better illustrated with 2-dimensional plots, by setting h_r or d while the other parameter remains a variable.

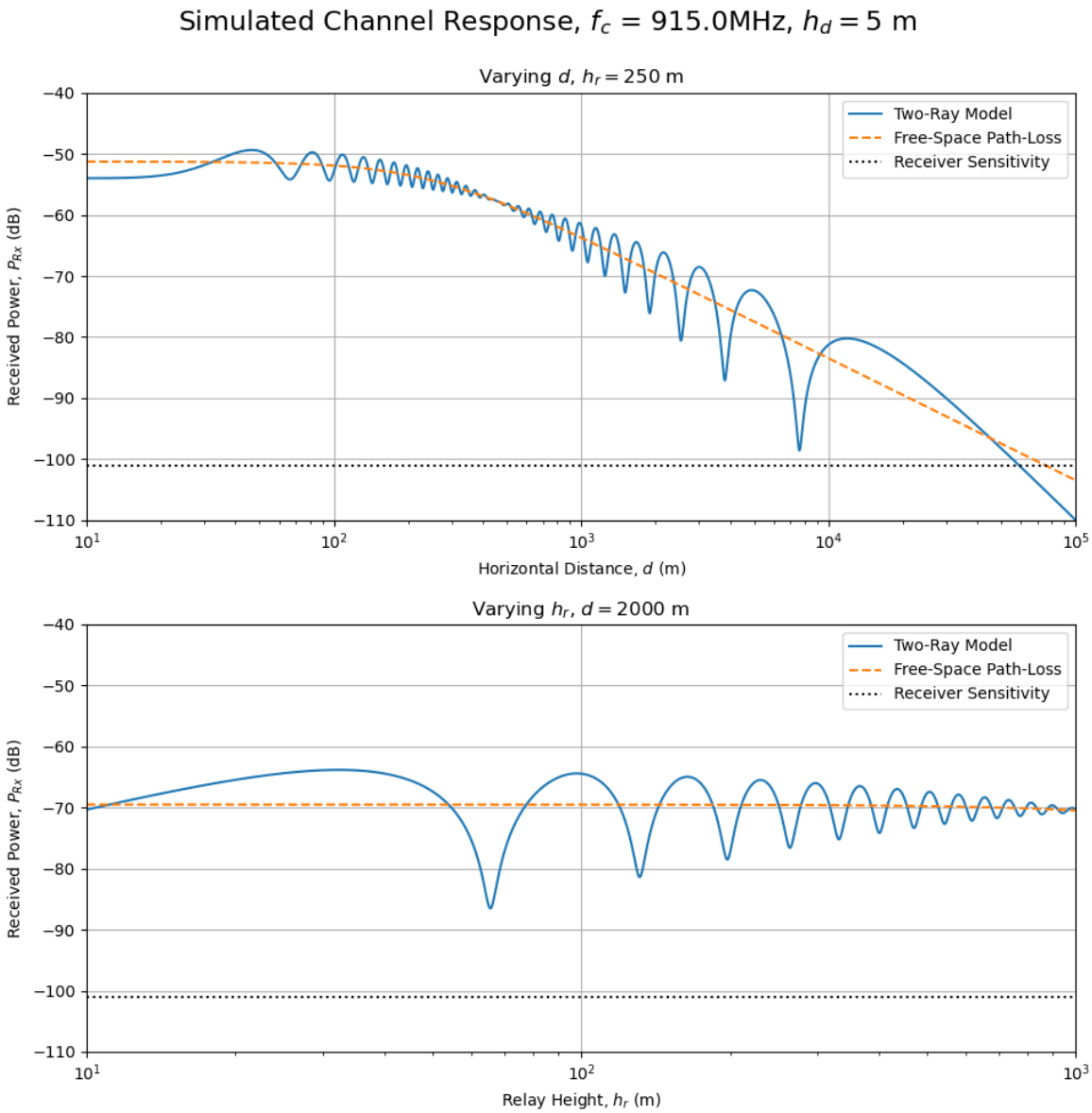


Figure 4.6: Cross-section of the two ray-channel model simulation seen in 4.5 and FSPL. The first plot has a constant $h_r = 250$ m and variable d, and the second plot has a constant $d = 2000$ m and variably h_r.

Cross-sections of Figure 4.5 have been taken at $h_r = 250$ m and $d = 2000$ m and shown in Figure 4.6. This better illustrates that under simulation conditions local minima and maxima with respect to d and h_r exist. If the two-ray model effect is somewhat apparent in a practical setting, then the spatial position of the relay UAV can be locally optimised.

4.1.4 UAV Spatial Position Optimisation

The two-ray model effect shown in Figure 4.5 and Figure 4.6, indicate that the strength of the received signal is highly dependant on the spatial position of the Relay UAV. A sinusoidal effect can be seen due to the phase difference $\Delta\phi$ between the interfering LOS and non-LOS rays. This implies that a local maxima will occur when $\{\Delta\phi = 2k\pi, k \in \mathbb{Z}\}$, and a local minima will occur when $\{\Delta\phi = (2k+1)\pi, k \in \mathbb{Z}\}$.

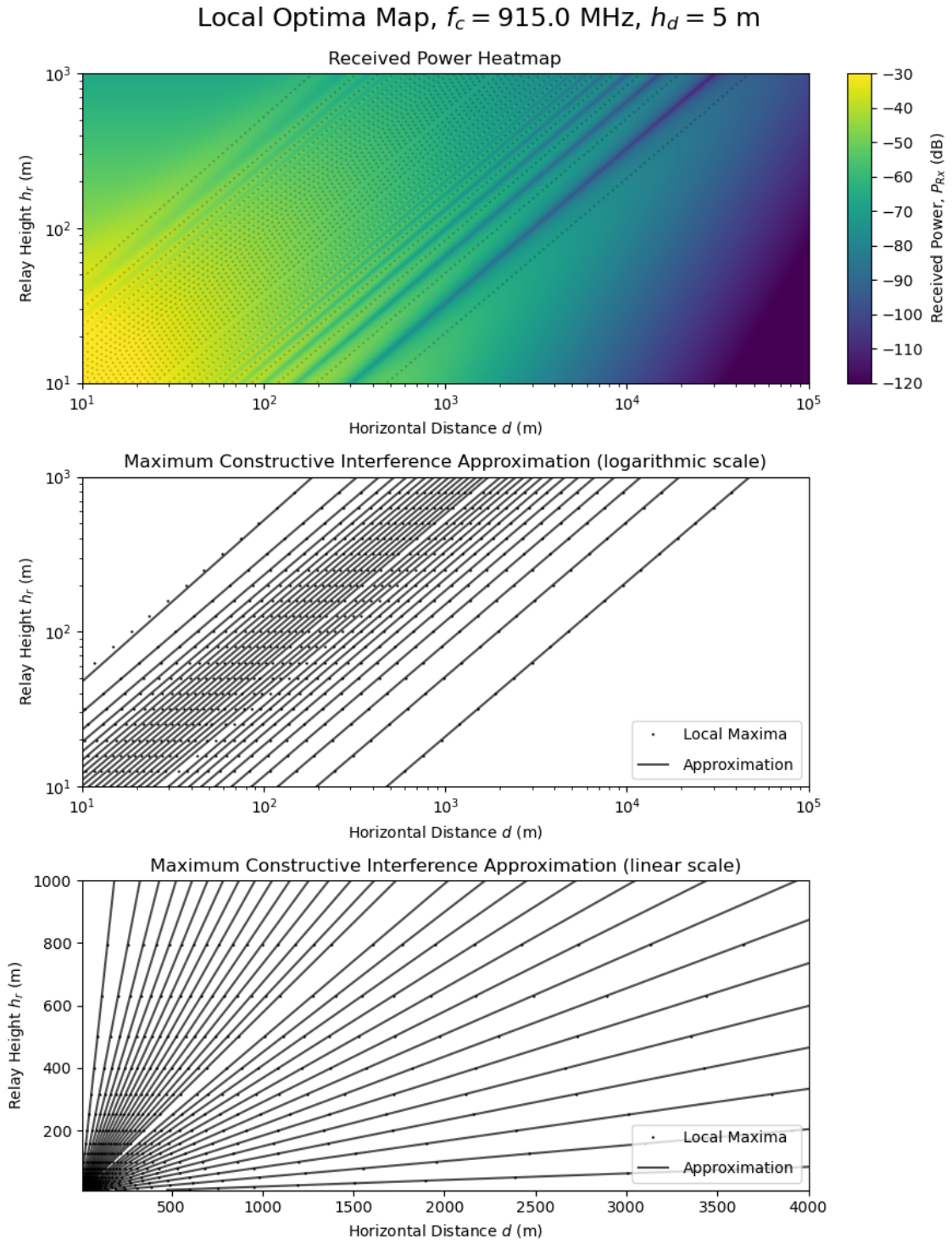


Figure 4.7: Heat-map of the two-ray model, as an alternative illustration of received power P_{RX} shown in Figure 4.5, with dotted lines superimposed on the top graph to indicate local optima.

Alternatively, this can be represented as a difference between the length of the LOS and non-LOS path with respect to carrier wavelength. Local maxima when $\{l - (r + r') = k\lambda, k \in \mathbb{Z}\}$, and local minima when $\{l - (r + r') = (2k + 1)\lambda, k \in \mathbb{Z}\}$. This concept is illustrated in Figure 4.7, where all local maxima were found by setting h_r and varying d. Dots represent the local maxima for cross-sections taken at set h_r intervals, and the lines are a linear approximation for each of the sinusoidal peaks.

The findings from this simulation imply that the spatial position of the relay UAV can be locally optimised for P_{RX}, if d and h_r lie on one of the linear approximations shown in Figure 4.7. However, applying this local optimisation in practise is not as simple as the simulation implies. This primarily comes down to two unrealistic assumptions about the reflective snow/ice surface, made to simplify the simulation. Firstly, in reality the snow/ice surface will not be perfectly flat and level, meaning that there may be multiple non-LOS reflections, each of which with a different and unpredictable $r + r'$ and ϕ_i. A statistical model can likely be applied to the reflective surface in order to more realistically model the multi-path propagation of the channel. Secondly, ϵ_r will vary with snow depth and other factors, rather than remaining a constant value of 3.17. Nullification of these assumptions implies that spatial optimisation of the inter-UAV communications link cannot be achieved without the aid of empirical data.

It should be noted that the spatial constraints of using antenna tracking directional antenna. This is shown in Figure 4.2, also relying on the assumption that the reflective surface is flat and level. While the antenna tracking constraints will differ in practise, the constraints of a directional antenna with fixed elevation angle $\phi_e = \frac{\phi_{bw,v}}{2}$ do not depend on this assumption. However, the specified directional antenna constraints rely on two unrealistic assumptions: uniform gain for $|\phi_{az}| \leq \frac{\phi_{bw,v}}{2}$, and negligible emissions when $|\phi_{az}| > \frac{\phi_{bw,v}}{2}$. Each of these can be accounted for by using an empirical model of antenna emissions as a function of ϕ_{az} in place of the aforementioned assumptions.

The same should also be done for the antennas used for the inter-UAV link, as a constant gain was assumed for all values of ϕ_{az}. This is not realistic when ϕ_{az} or ϕ_i approach $\pm 90°$ with standard length dipole or monopole antennas. Perhaps the application of an antennas emission pattern could be used to mitigate the effects of multi-path propagation for the inter-UAV link. However, further analysis has not been carried out on this subject, and did not lie within the aim or scope of this thesis.

4.2 Empirically Measuring Channel Characteristics

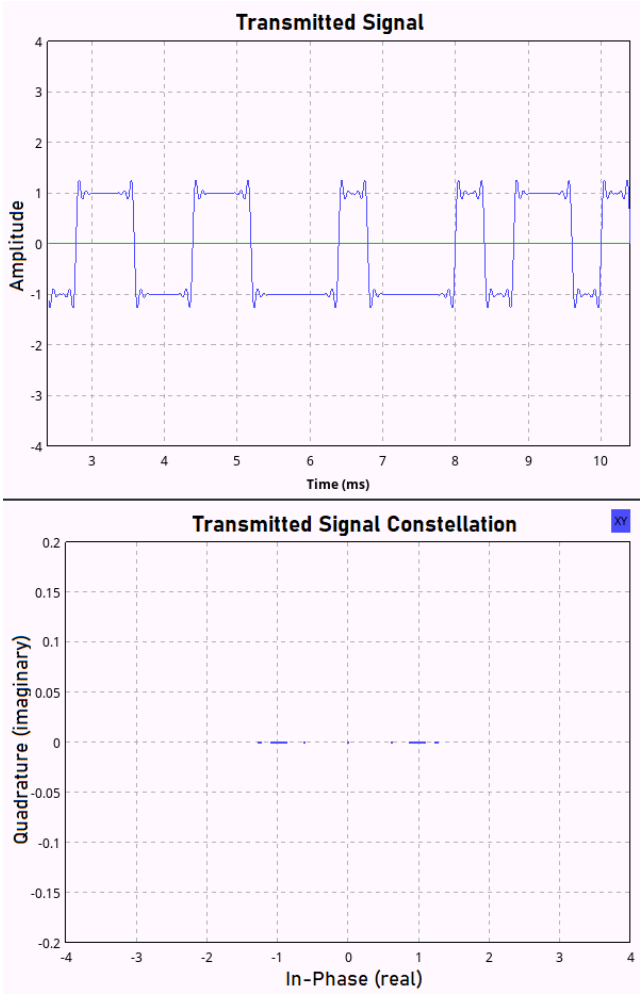
While simulating the channel model can provide insight into operational constraints of BVLOS missions prior to Antarctic deployment, simulations will inevitably differ from reality. Since this is the case, acquiring empirical data and generating a channel model will provide a clearer and more realistic insight into the channel characteristics over the Ross Sea. In addition, there may be an unforeseen relationship between the channel model and the snow thickness data being collected during missions. These reasons have prompted the development of a software defined radio (SDR) data collection system that can be used to approximate an empirical channel model.

Two Universal Software Radio Peripheral (USRP) radios were used for this prototype, as they support high data-rates up to 32 million samples per second (sps) with professional grade hardware. These SDRs are commonly used for research and development with a tool such as GNURadio for rapid development. The transmitter's software constantly repeats the same MLS, which is then BPSK modulated prior to transmission. The prototype data collection system was developed such that no SDR expertise is required. However, due to development time constraints and the need for simple deployment, this implementation was limited to collecting data without real-time signal processing. Therefore, the receiving configuration would obtain samples with the carrier signal removed, then store samples directly to a physical storage device for post-processing.

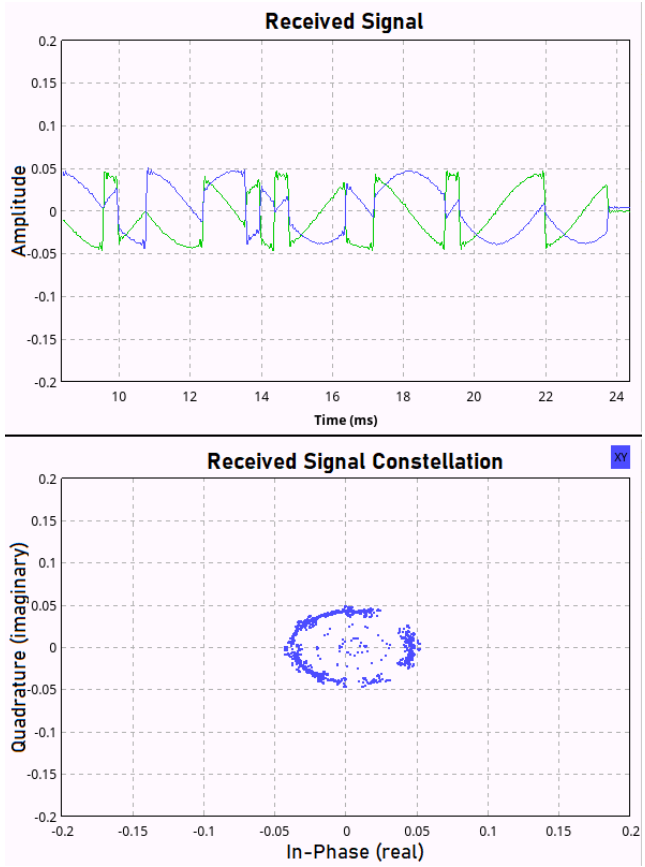
Carrier frequency offset (CFO) correction is a non-trivial process with SDRs, as the radios are designed such that desired receiver centre frequency is entirely software defined with no automatic hardware correction. Typical radio receivers can either tolerate up to a threshold for CFO offset, or alternatively will automatically correct CFO with a hardware feedback loop, such as a Costas loop. While GNURadio indeed has a software implementation of a Costas loop, there is no feedback to the SDR of any sort, meaning that the processing is entirely focused on carrier and phase recovery. Feedback loops in general are awkward to implement in GNURadio, and often require code to be written inside of the software script that is automatically generated from a flow-graph.

4.2.1 Carrier Frequency Offset Correction

In the later stages of prototyping, issues relating to received signal quality were identified. Analysis of the received signal revealed that a CFO was present, resulting in an undesirable, low-frequency sinusoidal effect in the received signal. The USRP b200-mini data-sheet states that the device has a frequency accuracy of ± 2.0 parts per million (ppm) [47]. This implies that signal transmitted at 915 MHz may have an offset within ± 1830 Hz. Since this frequency accuracy applies to both transmitter and receiver, the total CFO could lie within ± 4.0 ppm. Further complicating the matter, even if the CFO is manually corrected during operation, it appears to drift over time and vary with operation temperature.



(a) Transmitted signal.



(b) Received signal.

Figure 4.8: Comparison of the same radio signal before and after transmission with carrier frequency $f_c = 915$ MHz. The top plot on each shows the real (blue) and imaginary (green) components of the signal in real time, and the bottom plot depicts the signal constellation (real plotted against imaginary).

Figure 4.8b illustrates the effect CFO can have on received signal integrity. The initial solution to this problem used post-processing techniques to minimise the effect of CFO, using a Costas loop, digital filters, clock recovery and various synchronisation blocks in GNURadio. However, every time a digital signal is modified through each block's algorithm, information is lost due to the finite precision and artefacts can be introduced that do not truly reflect the original signal. Therefore, signal recovery that relies entirely on post-processing is less than desirable for channel sounding, which involves high-resolution correlations between the received and expected signals. As a result, efforts were made to limit the number of signal processing blocks and instead to account for the CFO automatically by adjusting the receiver frequency while collecting data.

The Costas loop block can be used in GNURadio to help minimise the effects of CFO and constant phase offset. However, it became apparent that the Costas loop software block will only recover the desired signal when the CFO is sufficiently low, with this threshold decreasing

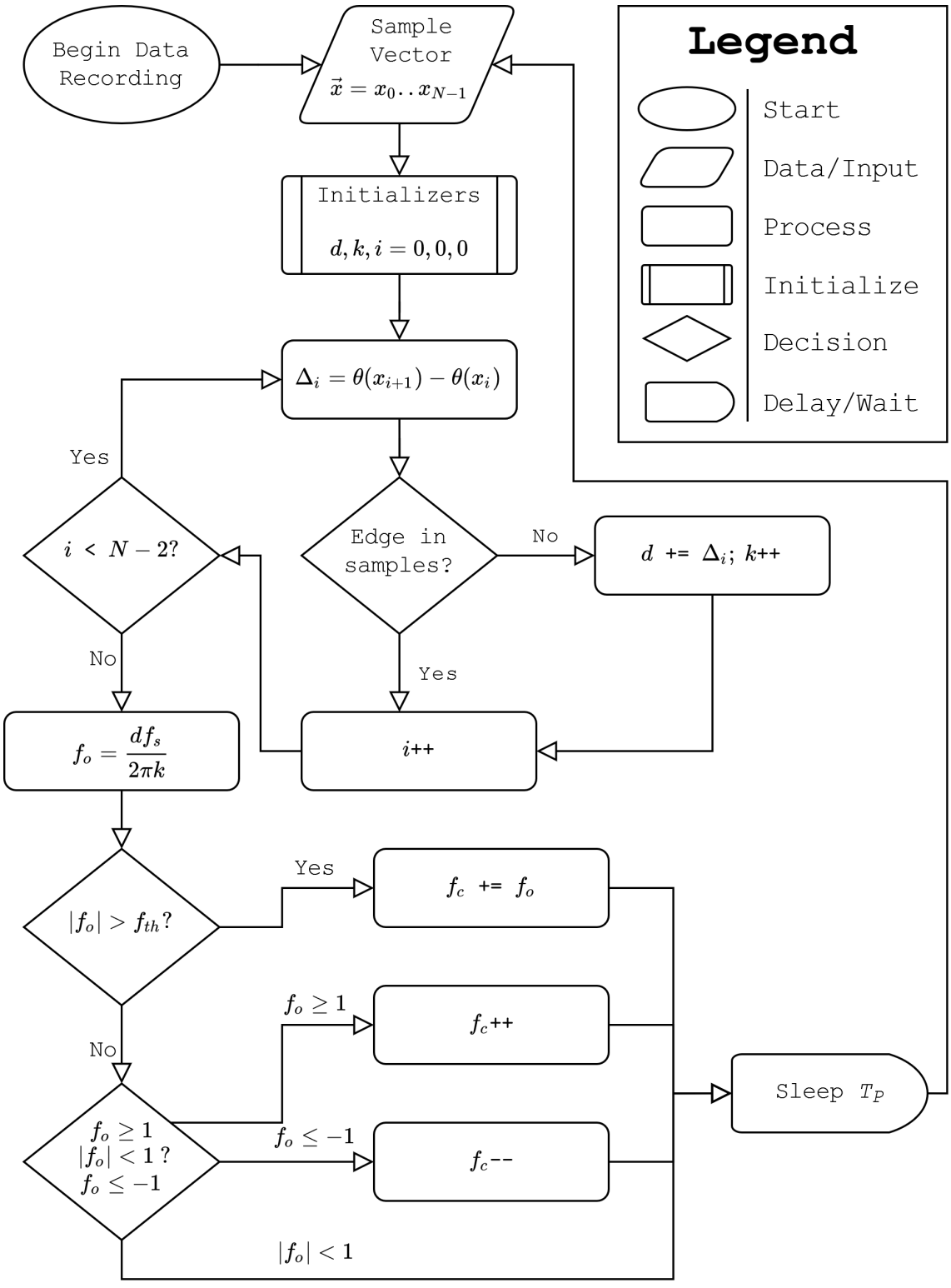
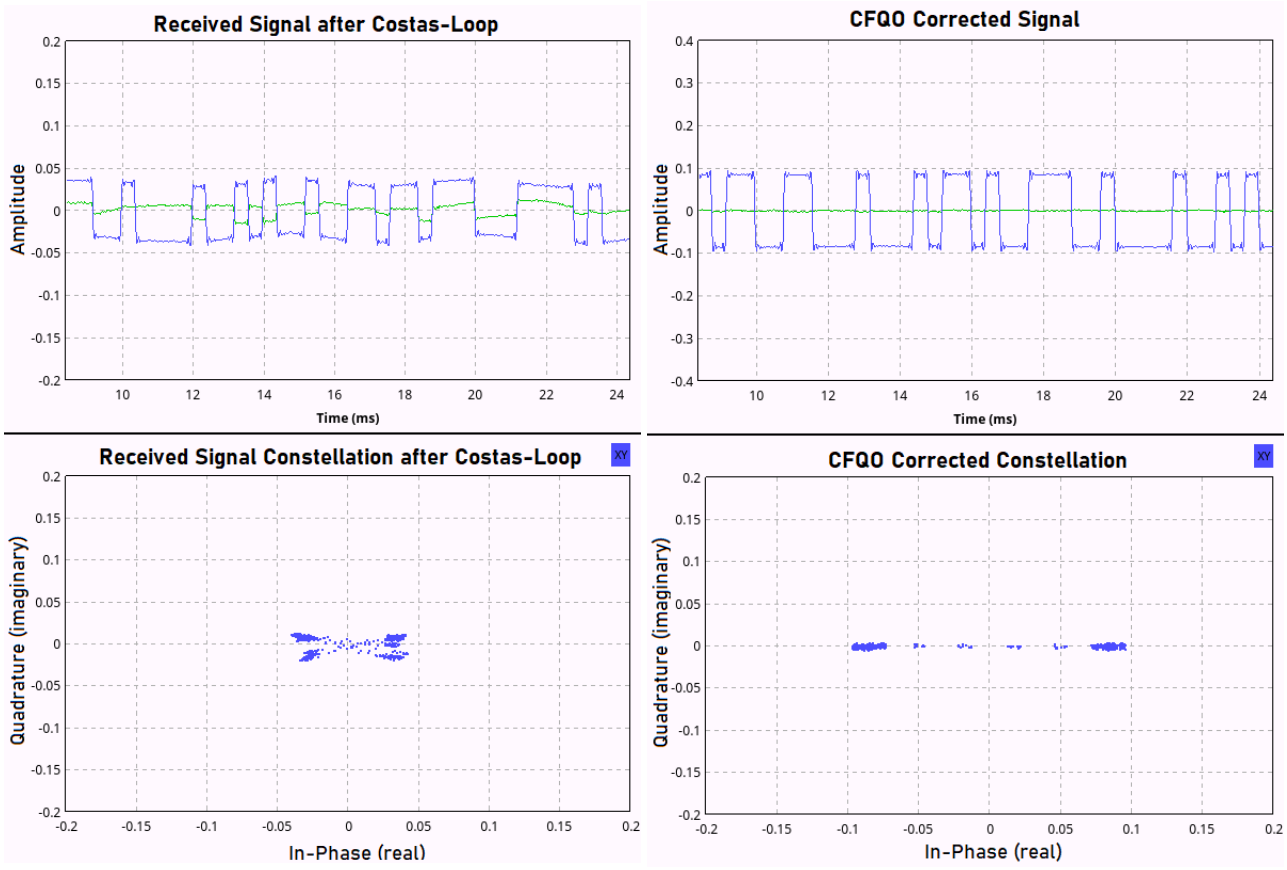


Figure 4.9: Flow diagram depicting how the frequency offset correction thread operates and interacts with the rest of the GNURadio script for data collection.

with signal strength. A piece of software was written within the auto-generated GNURadio script to calculate the CFO and adjust the USRP's centre frequency accordingly. For the best post-processing results, a higher sample rate will result in a smaller error. Each sample consists of two 32-bit floating point values, each representing the in-phase and quadrature amplitude of the sample. It was deemed a necessity for this additional process to have a negligible profile to

maximise sample rate without causing SDR buffer overflows. Trial and error revealed that 12 million sps (96 MB/s) was the maximum sample rate that could be practically achieved with field laptop with the data collection processes running. Samples are written without compression via USB 3.0 to a recording file, stored on external solid state drive.

The process shown in Figure 4.9 begins upon starting the GNURadio data recording script. A vector of N consecutive samples is probed for CFO estimation at the beginning each loop. Counters are initialised, then the process begins iteration over the sample vector $\vec{x}$. The phase difference Δ_i is calculated between consecutive samples x_{i+1} and x_i is calculated. To ensure the reliability of the algorithm, Δ_i is only added to d if consecutive samples do not lie on rising or falling edges of BPSK symbols. This is due to the phase change of samples on these edges, so instead of complicating the algorithm by accounting for them, they are ignored. If consecutive samples do not lie on an edge, then Δ_i is added to d and k is incremented. After iterating over $\vec{x}$, an estimate for the CFO, f_o, is calculated. How large the estimate of f_o is dictates the action taken to adjust the USRP's f_c. If f_o is larger than the threshold f_{th}, f_o is added to f_c. Otherwise, f_c is incremented if f_o is greater than 1 and decremented if f_o is less than 1, or remains unchanged otherwise. This final condition is due to the f_c resolution of the USRP being 1 Hz. Finally the process sleeps for T_P, then repeats.



(a) Costas loop output (no CFO adjustment).

(b) Costas loop output (autonomous CFO adjustment).

Figure 4.10: Comparison between the quality of recorded signals with (right) and without (left) the autonomous CFO correction script depicted in Figure 4.9.

Sample frequency f_s, as mentioned earlier is 12×10^6 sps. Vector length N was arbitrarily set at 1000 to ensure a large number of values were used to average f_o. Offset threshold f_{th} was set to 20 Hz, as the Costas loop block in GNURadio can correct sufficiently low CFO. Loop period T_D was set at a relatively large period of 1.0 s, as each correction loop is computationally expensive, and not many iterations are required to reach the condition $|f_o| \leq f_{th}$. The effectiveness of the automatic CFO method can be observed when comparing a sample of a uncorrected signal in Figure 4.10a to corrected one in Figure 4.10b. It should be noted that no training sequence is required for this method of CFO correction, as the phase difference of sequential IQ samples is all that is required.

4.2.2 Data Collection Plan

With a method tested and implemented to resolve the unexpected complication of CFO, the next stage was designing a flight plan for data collection. This is effectively a plan to observe the channel response as a function of d, at set values of h_r, similar to Figure 4.6.

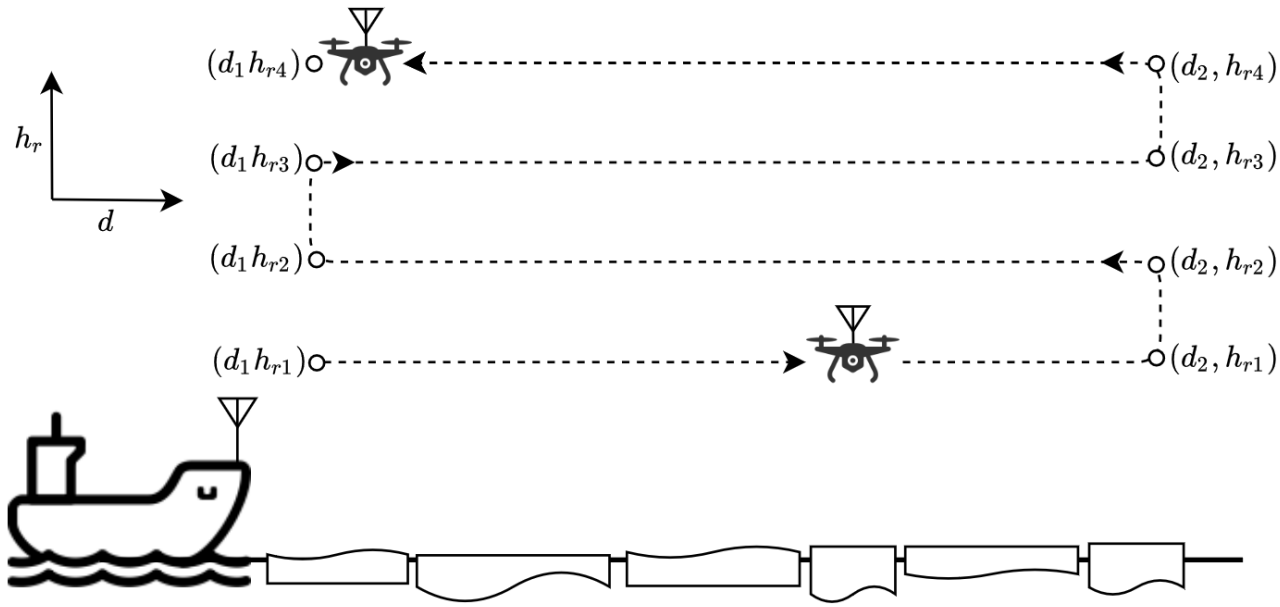


Figure 4.11: Spatial illustration of the data collection process, where data is continuously being recorded during the flight path.

A concise flight plan has been illustrated in Figure 4.11, where d will vary between d_1 m and the d_2. The latter being the maximum horizontal distance the pilot or supervising official is comfortable with, $d_2 = 1000$ m being and ideal value. Values of h_{r1}, h_{r2}, h_{r3}, and h_{r4} have been arbitrarily set to 25 m, 100 m, 175 m, and 250 m respectively. However, these can also be modified to fit any operational constraints imposed by the pilot or supervisor. Keeping to the arbitrarily defined values is not important, the main point of this test is to obtain raw signal data as d and h_r vary. This will then be post-processed to determine an empirical model for the channel as a function of the two variables.

4.2.3 Post-Processing

Equipment capable of performing the data collection tasks described throughout Section 4.2 had been prepared and sent to Antarctica for the 2018/2019 season. The initial plan was to collect data while on board an icebreaker during an expedition in the Ross Sea. Due to unforeseen circumstances involving a loss of equipment from another party, the expedition concluded prematurely, resulting in no field data being collected. Software to generate a power delay profile based on the data was in the early stages of development with the GNURadio toolbox. However, with no field data to work with, post-processing of this data was and designated as a future work and research endeavour. If a field data set is gathered in the future, analysis of the channel similar to that of the simulations seen in Figures 4.5 and 4.6 can feasibly be undertaken. This could be achieved by simply inferring P_{RX} from the data set, along with d and h_r from GPS coordinates, then plotting P_{RX} as a function of d and h_r. Another example of a secondary analysis from this data would be determining a loss metric such as bit error rate (BER), as a function of d and h_r.

4.3 Discussion

This chapter explored the wireless communications of BVLOS data collection missions on the physical layer. A directional antenna will be used as the GCS antenna, as it is physically viable, and would significantly mitigate multi-path reflections from the ice. As a result, the channel model becomes greatly simplified to that of FSPL, at the expense of spatial constraints, shown in Figure 4.2. Assuming FSPL and meeting spatial constraints, the maximum horizontal distance d_{max} of the URM2 link was calculated for each of MCS 0-15. This process was repeated, factoring in the ± 2 dB receiver tolerance of the URM2, with the lesser of the two d_{max} values at $P_{RS} + 2$ dB being used. These results were summarised in Table 4.2 and Table 4.3 respectively. Maintaining the range goal of 10 km would suggest that MCS 4 is the optimal choice, with a simulated range of 9.94 km (factoring in a +2 dB receiver tolerance) and network capacity of 39 Mbps.

With the set of simulation assumptions stated at the beginning of Subsection 4.1.1, the two-ray model was appropriate to model the wireless channel over the Ross Sea. A simulation of the two-ray model was written in Python, with graphical outputs shown in Figures 4.5 and 4.6. It was observed that under simulation conditions the spatial position of the relay UAV could be locally optimised for P_{RX} with respect to either d or h_r. These local minima and maxima are due to constructive and deconstructive interference, and these local optima are pronounced enough to make a significant impact on maximum operational range. For example, when $h_r = 250$ m, received signal power at $d = 2.5$ km was simulated to be -80.6 dBm, less than the received signal power of -80.2 dBm simulated at $d = 11.9$ km. Lines drawn in Figure 4.7 illustrate relationships between d and h_r where local optimisations occur. These optimisations are possible in theory, but rely heavily on simulation assumptions, and therefore, an

empirical model was suggested to determine whether spatial optimisation is possible in practise.

Consequently, a foundation was established for empirically measure channel characteristics with the use of SDRs and GNURadio. Unforeseen circumstances pushed the research and implementation of post-processing algorithms to become a topic of future work. Future research into this area should also look into the use of different modulation and spread spectrum techniques, as the foundation was designed around the transmitting and receiving of a BPSK modulated sequence. Furthermore, the performance of the CFO correction algorithm should be investigated for QPSK and M-QAM modulated signals.

Chapter 5

Final BVLOS Design & Future Work

Following the PoC design of Chapter 3, this Chapter describes improvements and additional features to enable reliable control of many UAV's for BVLOS data collection. Limitations with the original design were highlighted at the end of Chapter 3 and are subject to some of the improvements described in this Chapter. For example, bottle-necking of the DigiMesh network between the relay UAV and GCS was identified as one of the largest limitations of the previous design, with the network struggling with just two data collection UAVs and a relay UAV communicating with the GCS. Therefore, the wireless network is to be completely overhauled, with numerous optimisations to support more data collection UAVs with improved reliability.

5.1 Improving Network Architecture

After reviewing the PoC design implemented in Chapter 3, it was found that a purely DigiMesh network was not sufficient for a BVLOS data collection mission, involving the deployment of multiple data collector UAVs and one relay UAV. A diagram of the revised design to combat this bottle-neck, with a 2.4 GHz link between GCS and Relay UAV, is illustrated in Figure 5.1.

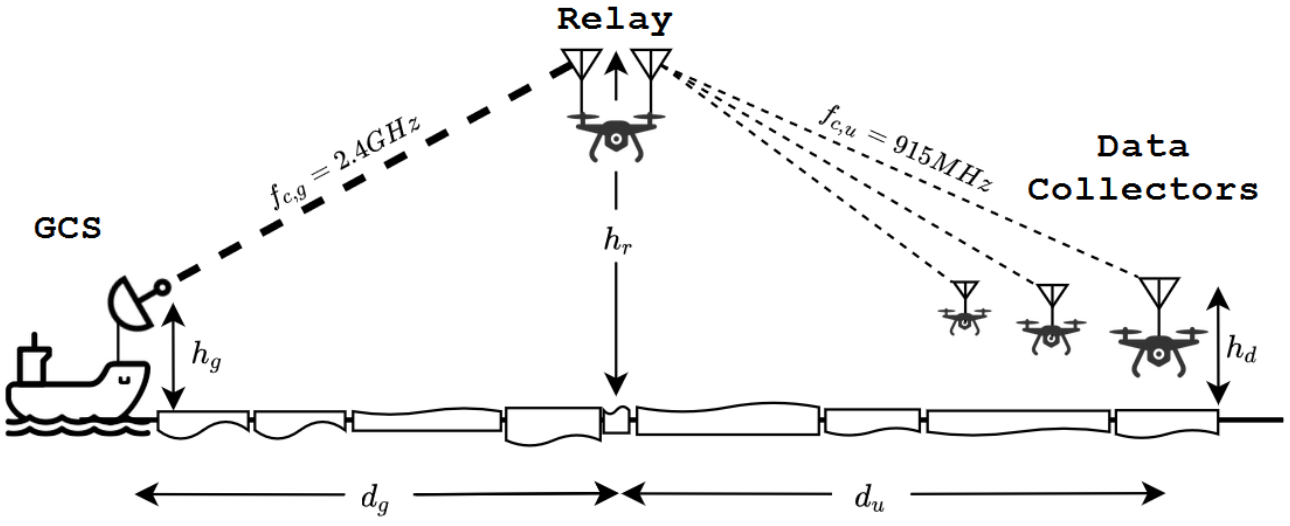


Figure 5.1: Spatial illustration of BVLOS missions, with key physical parameters labelled.

5.1.1 Eliminating the Bottleneck

The primary weakness from the PoC design was the network bottle-necking between the GCS and Relay transceivers. DigiMesh networks use frequency hopping spread spectrum (FHSS), and only allow for one device on the network to transmit at any given point in time. For a packet to be transmitted synchronously (awaiting acknowledgement) between a data collector UAV and the GCS, the packet must be received by the relay and re-transmitted, with the same process for the acknowledgement. This was confirmed during observations made in Chapter 3: the maximum telemetry rate with $n = 2$ data collector UAVs, even with an optimistic assumption for $D_{DM} = 41$ kbps, was calculated to be $D_{TELEM} = 8200$ bps. Therefore, from investigating the performance of DigiMesh for BVLOS data collection missions, it became apparent that a higher bandwidth link was required between the GCS and relay UAV.

This prompted revisiting the collection of RF hardware readily available, and exploring transceivers commonly among the open-source UAV community. Among the RF equipment available was the Ubiquiti Rocket M2 (URM2), a 2.4GHz transceiver supporting IEEE 802.11n MAC and PHY for creating long-range WLAN networks. The low-range often associated with similar WiFi devices is due to the noise and cluttered environment, paired with the weak transmit power of portable devices that are often equipped with compact and low gain antennas. Due to the lack of physical obstructions anticipated with the intended spatial configuration BVLOS missions, and the lack of other interfering RF devices, channel losses are anticipated to be far less than that of an urban environment. What sets the URM2 apart is the highly configurable settings depending on the desired application. Additionally, as mentioned in Subsection 4.1.2, the URM2 boasts a transmit power of 28dBm. In contrast, a WiFi router typically has a transmit power of 20dBm, and smartphones are typically limited to a transmit power of 15dBm with a lower antenna gain.

With the URM2 replacing the link between GCS and relay, the DigiMesh network no longer needs to transmit each packet between twice to exchange data between GCS and data collector UAVs. This is due to one transmission occurring on the DigiMesh network, and another on the 802.11n network. The bottleneck constrained BVLOS DigiMesh network described with Equation 3.1, can therefore be revised as

$$D_{TELEM} = \frac{D_{DM}}{n} \tag{5.1}$$

for this new configuration. Using the simpler telemetry rate formula, made possible by eliminating the GCS-Relay link bottleneck, we would have $D_{TELEM} = 20.5$ kbps for the $n = 2$ and $D_{DM} = 41$ kbps case. Applying the more conservative $D_{DM} = 25$ kbps results in $D_{TELEM} = 12.5$ kbps, a 150% improvement on the previous implementation that is bound by Equation 3.1.

Setting 802.11n parameters to maximise range, minimising data-rate (MCS 0, $B = 20$ MHz,

$T_{GI} = T_{DFT}/4$, and $N_{SS} = 1$), using Equation 2.6 results in $D_{MCS} = 6.5$ Mbps. This far exceeds what is required for merely maintaining multiple telemetry links, and could even potentially support additional data feedback. Two such forms of non-essential data would include sensor data from the snow radar, and images from each UAV, although individual data collector UAV links are limited by Equation 5.1. Sensor information could feasibly be integrated into a small, proprietary MAVLink message definition, slightly increasing the telemetry rate specified in Table 3.1. However, the transmission of images could not be supported, as the network capacity of DigiMesh would limit this process.

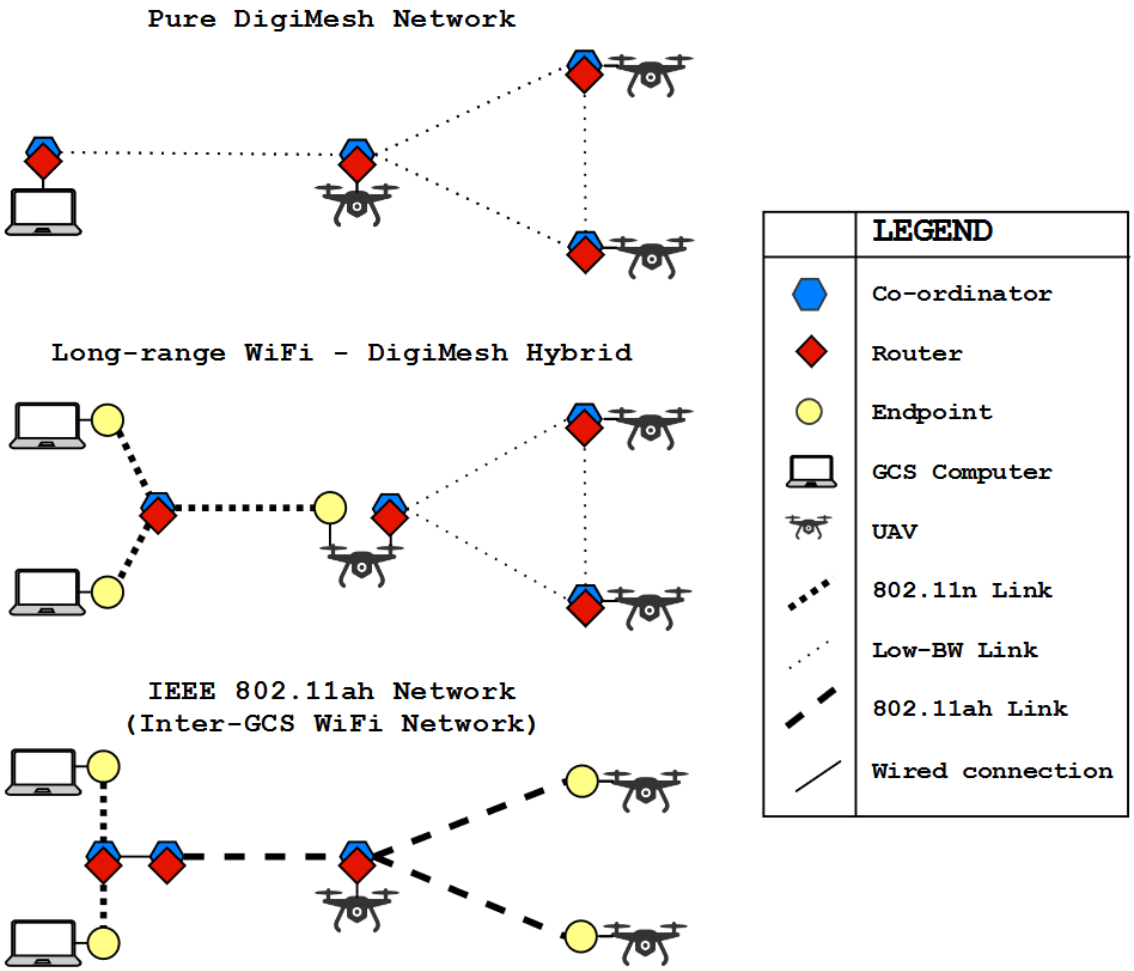


Figure 5.2: Comparison of network implementations for the PoC (top), network topology of the current design (middle), and future work topology recommendation using IEEE 802.11ah (bottom).

5.2 Modular Overview

Implementing long-range WiFi transceivers for communicating between the relay UAV and GCS vastly simplifies the configuration of the GCS. An URM2 is interfaced to both the relay UAV's companion computer, and to WiFi router for the GCS. The GCS URM2 is configured as a wireless bridge, physically connected to a WiFi router via a Power over Ethernet (PoE) connection. This enables devices that are wirelessly connected to the WiFi router to coexist

on the same network as the relay UAV, wirelessly connected to the GCS URM2. The relay URM2 is configured as a station (STA) and connected to the Intel NUC over a PoE connection. Upon supplying power to the relay UAV, the URM2 will boot and connect to the GCS network, as it has been automatically configured to do so in firmware settings. A comparison between the PoC topology and new network topology can be seen in Figure 5.2 as the top and middle topology illustrations respectively.

5.2.1 GCS Configuration

The modular configuration of the GCS is simplified to that of a more conventional configuration with no bespoke middleware required, as seen below in Figure 5.3.

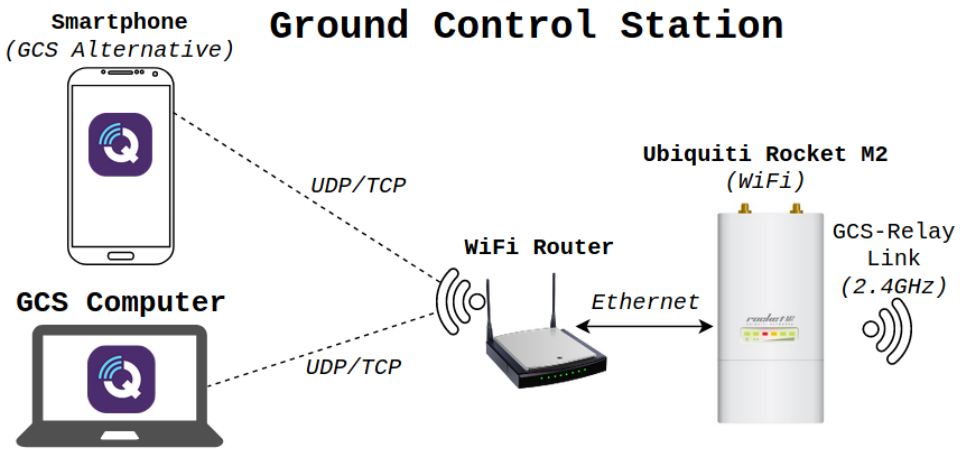


Figure 5.3: Modular overview of the final GCS configuration. Significant changes made since the PoC, most notably the GCS middleware has been removed, shown previously in Figure 3.3.

Any WiFi capable device compatible with QGroundControl can feasibly be used to pilot all UAVs in a BVLOS data collection mission. Similar to the PoC, each UAV is assigned a unique UDP port. However, in this design iteration any device on the WiFi network can establish a GCS connection to a UAV, rather just the device running GCS middleware. This allows for m pilots (each with a device) to share the responsibility of overseeing a data collection mission consisting of n UAVs, where $m \leq n$, removing prior limitation of one device being used to issue commands to all UAVs in a mission. It should be noted that the current implementation permits a UAV to maintain telemetry with only one GCS.

5.2.2 Relay UAV

With the GCS middleware being removed from the GCS device, in this design the Relay UAV is now responsible for bridging communications between the coordinator XBee and QGroundControl. Modifications have been made to both the GCS and flight controller middleware files, such that command-line arguments indicate what type of device each script is executing on. However, these modifications were made for companion computers accessible on the same WLAN as the GCS, such as the relay UAV as mentioned previously.

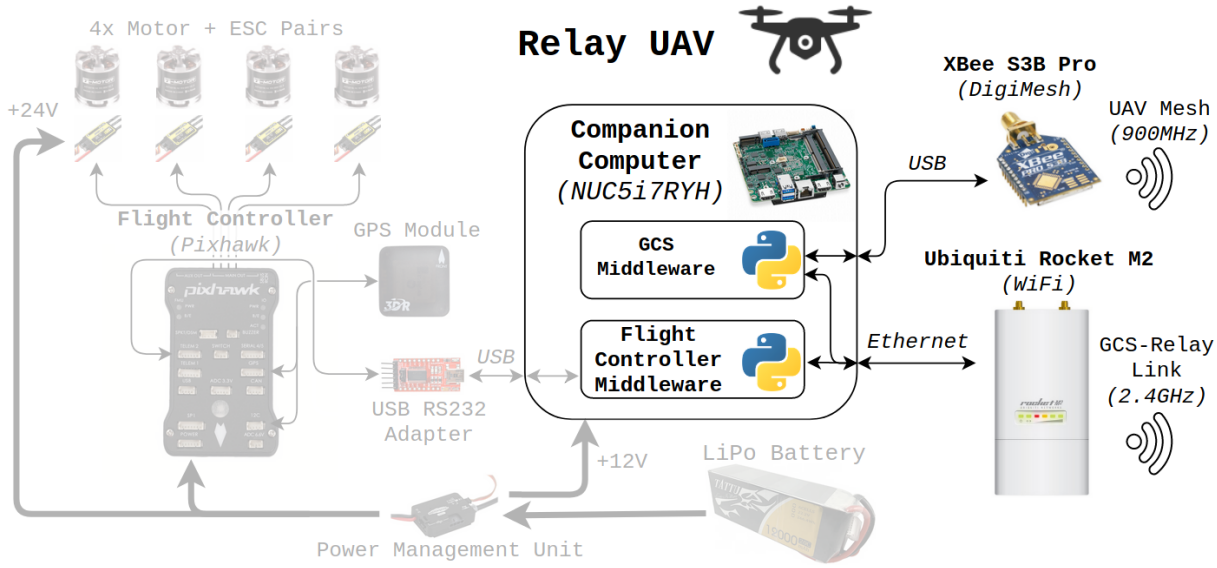


Figure 5.4: Modular overview of a Relay NAVI UAV, with emphasis on the changes made since the PoC modular design, shown previously in Figure 3.4.

5.2.3 Data Collector UAV

No functional additions or removals have been made to the data collector UAV configuration, described in Chapter 3 and illustrated in Figure 3.4. The only notable difference is the increased D_{TELEM} from removing the DigiMesh bottle-necking issue.

5.3 Future Work

5.3.1 IEEE 802.11ah for BVLOS Networking

Given the importance placed on reliability for Antarctic UAV missions, efforts should be made to mitigate factors that may compromise the reliability of BVLOS missions. Despite removing the network bottle-necking constraint of the PoC design, DigiMesh still limits operations with respect to data throughput. DigiMesh offered advantages in the form of a self-forming and self-healing network topology with automatic routing. However, this flexibility and simplicity was outweighed by the bandwidth constraints imposed.

Opting to extend the WLAN to include the inter-UAV network would provide both reliability and simplicity advantages over DigiMesh. For example, if each UAVs companion computer is directly IP addressable on the WLAN from the GCS, pilots can establish and maintain an SSH connection to the device. Also, direct UDP/TCP links between GCS and data collection UAVs can be established, eliminating the need for the GCS middleware that effectively acts as a bridge between DigiMesh and GCS WLAN. Finally, if the network capacity is increased to the point where rate limitation is not required on telemetry links, then the bespoke flight controller middleware could also be replaced. Open-source feature-rich alternatives such as MAVProxy [48] and MAVLink Router [49] are two such options for quick deployment of reliable transport

layer routing between flight controllers and GCS.

One alternative considered was the Ubiquiti Rocket M9 (URM9), a 900 MHz radio that utilises a 802.11n MAC and proprietary modification to the 802.11n PHY. Being from the same series as the URM2, the URM9 would provide the same features and hardware interfacing for rapid integration into the BVLOS system architecture. However, this model along with other Ubiquiti 900 MHz radios have been discontinued. It would be unfavourable to recommend incorporating legacy/discontinued hardware into future system designs, so other more modern alternatives will be investigated. Additionally, this hardware does not support any form of mesh-networking. All standards within the IEEE 802.11 family were investigated due to the aforementioned benefits associated with flying UAVs over an extended WLAN. Of the IEEE 802.11 family, standards 802.11p, 802.11s, and 802.11ah each offer features desirable for an inter-UAV network topology. RF hardware that supports the 802.11ah standard would provide the most suitable solution when compared to other IEEE 802.11 standards. Currently, the market for hardware supporting 802.11ah is limited, due to the early stages of the technology cycle. The 802.11ah standard is anticipated to become an important S1G standard for IoT applications due to it's mesh capabilities and improved range when compared to 2.4 GHz and 5 GHz RF modules. 802.11ah also supports much higher data-rate capabilities when compared to other S1G alternatives such as ZigBee, 6LoWPAN, and LoRaWAN.

To further simplify the BVLOS network architecture, 802.11ah compatible hardware could also replace the 802.11n compatible URM2s for the GCS – relay link. With this design choice, the system architecture would preserve the beneficial mesh networking aspects of DigiMesh: automatic message routing, self-forming network, self-healing network, trivial relay swapping, etc.. The introduction of the URM2's for the GCS – relay link solved the bottle-necking issue from the PoC, although relay swapping was no longer trivial with this implementation. Additionally, 802.11ah provides far greater network speeds and enables direct IP addressing, similar to that of any WLAN. For example, MCS 10 offers the smallest 802.11ah network capacity of 150 kbps, where $N_{SS} = 1$, $B = 1$ MHz, $T_{GI} = T_{DFT}/4$. This still far exceeds DigiMesh's theoretical maximum of 92 kbps, let alone the practical maximum of around 41 kbps. An analysis similar to that shown in Figure 4.3 and Table 4.2 would be useful in determining what MCS, T_{GI}, and channel bandwidth should be used. This would inevitably depend on the hardware on which the 802.11ah standard is implemented, so no such analysis has been carried out at this stage.

5.3.2 Additional Features

Network capacity increasing substantially through the use of 802.11ah or a similar technology would provide a foundation for useful features and enhanced reliability. An itemised list of useful features for BVLOS missions is as follows:

- **Image/Video Feedback and QGroundControl Integration**

  This is the least complicated of the suggested features, as it does not require any clever software implementation of hardware configuration. If the network capacity can support an image or video feed from each UAV, then a feed can be established with QGroundControl. A software tool is used to broadcast a camera's data over UDP, such that QGroundControl can connect to display the video feedback.

- **Mid-Mission Relay Swapping**

  The ability to swap the relay UAV during BVLOS missions improves the reliability of operations, and potentially can extend the duration of missions. One useful circumstance of relay swapping would arise if the relay UAV is faulty in some aspect. The entire data collection would typically be ended early, with an RTL command sent to all UAVs. Cancellation of the mission can potentially be avoided in this situation by deploying a second relay UAV, then sending an RTL command to recall the first. The other notable circumstance would be to extend the duration of missions. Future design iterations of the snow radar will likely yield a sensor that can physically be mounted upon a fixed-wing UAV. Fixed-wing UAVs cannot hover to maintain a fixed position in space like multi-rotor UAVs, but they are far more energy efficient when flying in a straight line at high velocity. This means fixed-wing UAVs would be more suited for data collection, whereas a multi-rotor UAV would remain suitable for the relay as a relatively constant position in space is maintained (slight variations if local position optimisation described in Subsection 4.1.4 is used). Therefore, if such a configuration is used for BVLOS data collection missions, then the ability to swap relays could increase data collection mission duration and consequently increase the size of the area surveyed.

  The implementation of relay swapping is trivial if the BVLOS network is that of a self-forming and automatic routing mesh network. With the PoC design a replacement relay UAV could be launched, reach a similar position in space to the original relay UAV, then a RTL command would be sent to the original relay. This simple process would ensure continuous telemetry links between data collection UAVs and the GCS. The final design provided improvements in the form of network capacity, although it compromised on the simplicity of the network and made the relay swapping process more complicated. This is due to GCS being configured as an access point, and relay UAVs being configured as stations that automatically connect to the GCS. Once the replacement relay UAV reaches the desired position, a handover broadcast must be sent to all data collector UAVs, causing data collector UAVs to send telemetry data to the new relay UAV's XBee address.

  In addition to this slightly more complicated procedure, since the new relay UAV has a different IP address the GCS must change IP address for each of the data collector UAVs connected. The unusual process of UAVs changing their IP address mid mission is

currently not supported by GCS clients, and therefore is assumed to be a completely new UAV. This implies the telemetry link would have to be re-initiated with each UAV, and discontinuity between mission logs. An additional software script operating on the GCS could be used to avoid the changing IP address issue, but this would inevitably increase the complexity of BVLOS data collection missions. Replacing the hybrid BVLOS network with 802.11ah or similar technology, would make the relay swapping process trivial once again.

- **ROS/MAVROS Companion Computer Integration**

  ROS has become a commonly used technology for both academia and industry for on-board decision making of robots and UAVs. The bespoke design of the flight controller middleware currently does not support ROS interfacing and expects a direct serial connection to the flight controller. This implies no support for other on-board computing and autonomous decision making processes. Examples include obstacle detection, collision avoidance, mission re-mapping, and additional conditions to trigger RTL.

  One approach would be to use MAVLink Router to directly interface with the flight controller, and modifying the middleware script to expect a UDP connection with MAVLink Router. Since MAVLink Router can create multiple unique connections with the flight controller, ROS could be interfaced simultaneously with the middleware script through separate UDP links. This approach would retain the telemetry rate limiting functionality, while also enabling ROS interfacing for increasing the level of UAV autonomy.

- **Real-time Channel Measurements**

  Theoretically, if the two-ray model is an accurate approximation for the channel then local position optimisation of the relay UAV can be determined, as described in Subsection 4.1.4. Realistically the two-ray model will be an over-simplification of the channel, and any attempts made at position optimisation should only be made based off empirical measurements of the channel. Section 4.2 describes a basis for sampling the raw signal at the receiver (carrier frequency removed). It is feasible that the data collected could be processed to obtain channel characteristics. Combining this with reported radio metrics such as BER and RSSI, then mapping the data as a function of d and h_r could result in local position optimisation based on empirical data. A feasibility should be conducted on both real-time channel measurements, and position optimisation based on such measurements.

- **Autonomous Flight Enhancements: Obstacle Detection & Avoidance**

  This feature/improvement requires completion of the previous point: ROS integration on the companion computer. With this interface enabled, autonomous flight features and proprietary RTL logic can be added to both data collector and relay UAVs. While there may be very few, or perhaps no physical obstacles for data collection UAVs to avoid, autonomous features could still prove useful when adhering to regulations. For example,

as mentioned in Subsection 2.1.1, wildlife activity should be monitored by a VLOS observer in the case of changing behaviour [12]. This recommendation cannot be fulfilled once data collection UAVs have traversed a great distance, but on-board sensors and decision making software could aid in this matter. To avoid any complications from nearby wildlife, sensor information and recognition software could feasibly identify clusters, then consequently re-map the data collection flight plan to avoid disturbance.

- **Real-time Snow Radar Feedback**

  If the anticipated data-rate of the sensor is small enough, the snow radar could be treated as a flight sensor and data could be incorporated into the telemetry link. This can be achieved by interfacing the output of the snow radar sensor directly to the Pixhawk via an ADC or I2C, and ensuring the correct MAVLink message type is part of either the prioritised set of messages, or regularly scheduled set of messages. Taking this approach means that implementation of packet overhead, error correction, or cyclic redundancy checking is surplus to requirement. Snow radar data would be contained in the same log file as the mission, synchronising measurements with positional data. Automatic synchronisation of position and snow depth measurements would prove highly beneficial for combining the data feedback with satellite measurements.

  Measurement frequency, $T_S = 0.5$ s, raw measurement size of 3204 B [50], over 13 MAVLink packets would be required for transmission, due to the maximum payload size of 255 byte. This adds an additional 156 bytes of overhead to each measurement. Each telemetry link would need to be increased by 53.76 kbps for real-time feedback of snow radar measurements, for $T_S = 0.5$ s. Future iterations of the snow radar will support $T_S \leq 0.05$. At this point the transmission of raw data back to the GCS may be excessive and compromise the integrity of the telemetry link. As an alternative, data could be processed on board to obtain depth inferences, with those processed values being transmitted back to the GCS. If the processing time of raw measurements exceeds the period between samples ($T_P > T_S$), then data must be stored on-board or some kind of decimation must occur. Otherwise ($T_P \leq T_S$) a GPS coordinate reading should be taken as the measurement is taken. After processing, the depth value and GPS coordinate can be transmitted in a custom MAVLink message. Alternatively T_P can be measured then used to map readings to coordinates at the GCS.

Many other possibilities for future work to build upon this project exist, with the list above mentioning some obvious choices. Regardless of what additions are made, the most important goal for the future of this project is the original mission: collecting snow thickness data in Antarctica quickly and reliably over large areas of sea ice. The suggestions made for future work should be considered and some potentially prioritised, but should not be viewed as a list of pre-requisites for deployment of BVLOS data collection missions. Therefore, compromising on the implementation of less important features in order to make the deadline for a summer

expedition would be favourable, unless such features are deemed essential.

5.4 Discussion

An emphasis was placed on mitigating the bottle-necking effect of the PoC design, by migrating from a purely DigiMesh network to an 802.11n – DigiMesh hybrid. The outcome was an improvement to the total network capacity, simplification of the GCS configuration, and IP addressing of the relay UAV. A summary of the benefits and drawbacks of the changes made in this final design are as follows:

- **Benefits**
  - Mitigation of bottle-necking effect, the hybrid network can now support three data collector UAVs without lowering the default load of 10.93 kbps for each telemetry link, assuming $D_{DM} = 41$ kbps.
  - The MCS can be altered to maximise the data throughput while being mindful of range requirements. A comparison of theoretical range by MCS can be seen in Subsection 4.1.2.
  - Simplified GCS configuration, with relay UAV and middleware generated UDP links to each data collector UAV being accessible from any device on the local GCS WiFi network.
- **Drawbacks**
  - More complicated network architecture when compared to PoC design. PoC network was completely self-forming and self-healing, whereas the same is only true for the inter-UAV network in the final design.
  - URM2 is legacy hardware and no longer in production, and potentially beneficial airMAX features are non-compatible with other 802.11n devices.

IEEE 802.11ah was found to be the most ideal among the 802.11 family of open standards, due to it's mesh capabilities, IP addressable WLANs, and enhanced range through the use of S1G frequencies. An implementation of hardware supporting this standard was not pursued during this thesis, as the market for this technology is still in it's infancy. In the near future when the market has further matured with additional hardware options becoming available, implementing 802.11ah for BVLOS data collection missions should be a high priority. This would increase network capacity, provide the simplicity benefits associated with mesh networking, and improve the reliability of BVLOS missions with direct IP addressing of each UAV's companion computer. Consequently, this would open the door to the previously mentioned additional features in Subsection 5.3.2, such as real-time snow radar feedback, improved autonomy, and image/video feedback.

Chapter 6

Policy Challenges & Recommendations

While the clear focus of this project is on the problems posed by BVLOS UAV operation and the corresponding practical solutions, the current state of both New Zealand and Antarctic UAV policy will be taken into consideration when defining operational boundaries and specifications. Future UAV based projects will benefit from more lenient regulations around autonomous and BVLOS operations in New Zealand. Projects such as this may be used as as example for policy makers to modernise aerospace regulations in New Zealand, and further define Antarctic UAV recommendations. The aim of this chapter is to identify ambiguity in regulations and restrictions that limit operations such as the one outlined in this research, and then provide a clear and scientific basis for recommendations. Recommendations will be made to existing CAA rules for New Zealand UAV operations, and COMNAP guidelines for Antarctica UAV operations, specifically regarding different modes of flight and BVLOS operations.

6.1 Current Regulations & Limitations

Procedures and recommendations surrounding UAV use in Antarctica and New Zealand have been summarised in Subsections 2.1.1 and 2.2.3 respectively. Changes to rules around airspace may be imminent with the recent rise in interest of personal air-carriers and autonomous air-taxis, along with the increased interest around commercial UAV applications. CAA has demonstrated a willingness to support proven technology solutions for BVLOS operations [51, 52, 53], as they, along with Callaghan Innovation, recognise the economic potential of fewer restrictions around BVLOS [54].

In New Zealand, local authorities reserve the right to impose additional restrictions on UAV operations, particularly as as Part 101 requires permission from the land owner(s) to fly over. For example, the University of Canterbury's controlled airspace at Birdlings Flat has additional conditions placed upon its usage by the Christchurch City Council (CCC) and Department of Conservation (DOC). These restrictions included a minimum flying altitude of 50 m over wildlife hot-spots, along with a requirement that all flight logs must being submitted to the CCC post-operation, to ensure the conservation of vulnerable coastal species. To carry out field trials

that replicate the conditions of BVLOS data collection missions, data collection UAVs must maintain the minimum altitude of 50 m until sufficiently far from the coast, where the UAV's then descend to an altitude of 5 m to 15 m.

6.1.1 Ambiguity & Assumptions

New Zealand's UAV regulations are well defined and restrictive in most instances, but it is evident that current UAV policy has been developed around the manual mode of flight. Antarctica has very few enforceable regulations around UAV flight, and often relies on a competent authority from COMNAP to approve missions whether missions take place and collect flight logs. The following is a summary of the key takeaways from existing regulation that form the basis of this chapter:

- No distinction has been made among varying modes of UAV flight for either New Zealand or Antarctic regulations. It appears that most regulations have been designed around manual flight, where a pilot provides real-time movement commands to UAVs. However, this mode of flight differs significantly to a flight controller and autopilot software executing a mission map.

- Neither Antarctica or New Zealand regulations mention autonomous flight features. This may be considered as another mode of flight, but levels of autonomy and intelligence can vary greatly. For example, a UAV following a planned mission map may also have onboard sensors to identify obstructions that would trigger autonomous collision avoidance action. The quality and effectiveness of features such as collision avoidance depends on: the type of sensors and their precision, algorithm effectiveness, and processing delay.

- In New Zealand supervisors and pilots must maintain an unobstructed VLOS at all stages of flight. This can be achieved with this assistance of visual aid. A competent observer can maintain a VLOS instead of the pilot, assuming clear verbal communication between the pilot and observer. Antarctic BVLOS missions can be approved by a competent COMNAP authority, but is recommended (yet not required) that VLOS is maintained by a supervisor when possible.

- New Zealand regulations from the CAA state that direct wireless communications is required between the UAV and either GCS or RC. No mention is made of indirect wireless communications, nor are direct or indirect wireless communications defined in this context. It was assumed that control of a UAV via a telemetry link that requires multiple network hops for data transmission is prohibited outside of an activated restricted airspace. Therefore, BVLOS flight with relayed communications in New Zealand is prohibited, unless in an activated restricted airspace.

6.2 Recommendations

This section will put forth recommendations for both Antarctica and New Zealand regulations that may affect UAV related operations. These recommendations are provided with the intent of better defining constraints and guidelines for all manner of UAV operations.

6.2.1 Distinction Among Modes of Flight

As noted, there is currently no regulatory distinction between manually controlled UAVs, autopiloted UAVs following a flight map, and varying levels of on-board autonomy in UAV flight. Planned missions involve uploading a set of 3D co-ordinates and velocity vectors to the UAV, defining how the flight-controller will navigate the UAV without any manual input from a RC. No regulations applying to UAV use in either New Zealand or Antarctica have explicit rules for planned way-point based missions. From a regulatory standpoint, these modes of flight are treated in the same way as the manual mode of flight.

Autonomy in UAV flight also garnered no mention in Antarctica or New Zealand UAV documentation. Defining a set of regulations for autonomous flight is complicated, due the variation in quality and level of autonomy. It could be argued that autonomy is core to modern UAV functionality with the features provided by autopilot, e.g. position holding and the ability to navigate a set of way-points without guidance from a pilot. Flight controllers and autopilot software provide the basis for planned UAV flights and assistance to manual flight. Therefore, to better clarify these recommendations autopilot will be considered a core feature for all modes of flight:

- **Manually Piloted Flight**
  In a manually controlled flight mode, the pilot provides real-time movement commands to the UAV during operation. It appears that the majority of UAV regulations have been written with manual UAV flight as the only application. As a result, no recommendations have been made towards the regulations around manual UAV flight, other than clarification as to which rules are unique to manual flight if a distinction is made in future iterations of UAV policy.

- **Planned Missions**
  An evaluation of the environment to determine any anticipated hazards can determine whether autonomous features are required to reliably execute the mission. Using BVLOS data collection missions in the Ross Sea, it can be reliably assumed that physical obstructions will block the planned path of data collection UAVs at an altitude of 10 m. However, this relies on all sensors and subsystems of the UAV operating as intended for the flight path to be followed with accuracy.

- **Autonomous Flight**
  Research surrounding autonomy in flight was not within the scope of this thesis. However,

from a high-level overview it became clear that autonomous flight should be treated differently to manual flight in regulations. "Sense and avoid" is a common autonomous feature that assists in planned missions, making slight real-time alterations to the flight path based on environmental hazards. In advanced applications, a flight path will be generated based on a specific task defined in software, displaying a different level of autonomy than flight assistance features.

6.2.2 Data Requirements

Within the results described in Section 3.5.1, MAVLink message types that constitute a standard telemetry link were summarised in Table 3.1. It was noted that redundancy existed within some of the messages transmitted to the GCS from UAVs. After this was observed, it was then speculated that perhaps a minimal set of messages could be defined for BVLOS operations. This minimal set, that has redundant information removed, could act as a benchmark, and in addition reduce the D_{TELEM} requirement.

Redundant MAVLink Message Types	
MAVLink ID	**Rationale for Redundancy**
RC_CHANNELS	RC not present for BVLOS missions.
RC_CHANNELS_OVERRIDE	RC not present for BVLOS missions.
ODOMETRY	Odometry is a collection of state information found in other MAVLink messages types that are compliant with ROS interfaces. If ROS integration is perused, the odometry message will by handled by the companion computer, not transmitted back to GCS.
SERVO_OUTPUT_RAW	No servos used in the existing UAV configuration for either relay or data collector.
GPS_RAW_INT	Raw GPS sensor data. A more useful position estimate is given in GLOBAL_POSITION_INT.
LOCAL_POSITION_NED	Filtered local position obtained through sensor fusion of GPS, computer vision, and other similar positional sensors. Local position more useful for ROS interfacing than GCS, which uses GLOBAL_POSITION_INT for graphical representation of position.

Table 6.1: Redundant MAVLink message types and the rationale for their redundancy.

A summary of redundant MAVLink message types, along with a reasons as to why they are

considered redundant for a minimal telemetry link is shown in Table 6.1. The two RC message types had already been ignored in Section 3.5.1 when calculating the standard telemetry rate, yielding a value of 11.10 kbps. Ignoring the other message types mentioned in Table 6.1 lowers this by 35%, to 7.19 kbps. It could also be argued that more of these MAVLink message types are not required for a telemetry link, because the information enclosed may not be deemed essential. In addition to the removal of redundant message types, the period between the transmission of each MAVLink message type could also be considered to establish a minimum benchmark for BVLOS flight. No exact values will be put forth in this thesis for such transmission periods. However, it is recommended that further investigation is carried out into a minimal set of MAVLink messages and corresponding transmission periods, as it could lead to explicit telemetry data requirements being defined in regulatory documents.

6.2.3 Clarity on BVLOS Regulations

If the suggestions put forth in Subsections 6.2.1 and 6.2.2 are addressed, leading to definitions on modes of flight and data requirements, then existing BVLOS restrictions can be re-considered. Since manually piloted BVLOS flight is not practical[1], distinction among modes of flight in policy is a pre-requisite to reconsidering existing BVLOS regulations.

Antarctica would benefit from clearer guidelines on BVLOS flight, as currently a decision is made by a COMNAP authority based on risk associated with the mission. Clear guidelines for the different modes of flight, and explicit data requirements to assist COMNAP authorities in determining whether a mission can be carried out safely and reliably in accordance to their own recommendations. For example, if a UAV was manually flown and the pilot or observer notices a colony of penguins nearby, then it is the pilots duty to maintain an agreeable distance such that the colony is not disturbed. A UAV could feasibly demonstrate the ability to autonomously detect wildlife, then alter the flight path accordingly to maintain a set distance. This would result in a higher level of confidence associated with unsupervised flight with such autonomous features. If data collection UAVs equipped with snow radars can demonstrate this ability, then COMNAP can have greater confidence that BVLOS data collection missions will not disturb wildlife. Alternatively, an example of a data requirement may involve COMNAP suggesting real-time FPV video feedback from each UAV in a BVLOS mission.

If BVLOS is to become permissible outside of restricted airspace in New Zealand, some kind of maximum range must be defined. Research and discussion between CAA, engineers involved in the UAV industry, and policy makers may lead to an explicitly defined range [53]. However, an alternative put forth in this thesis will be the integrity of the telemetry link between a UAV and the GCS where VLOS is not present. A moving average for packet loss along with a threshold percentage would provide the basis for a dynamic maximum range. Packet loss is determined

[1]Possible exception for trained pilots utilising a first-person-view (FPV) camera.

within the GCS and is therefore independent of radio hardware or network protocol used. For example, if a 5 second moving average for packet loss rises above 50% a RTL command could be automatically triggered within QGroundControl and sent to the UAV. An analysis should be carried out to define explicit values for: duration of the moving average, packet loss threshold, and any other useful link quality factors.

6.3 GCS Software Integration of Regulations

Modern open-source GCS applications such as QGroundControl and MissionPlanner support a variety of tools to assist in planning reliable missions that adhere to regulations. Examples of such features include rally points (alternative RTL locations) and GeoFence (imposed flight boundaries). Currently, GCS applications have no direct support for UAV networks, and assume that a PtP link is being maintained between each UAV and the GCS. With rapidly evolving technology, growing number of applications within the UAV landscape, and discussion about MAVLink version 3 indicate that the open-source UAV ecosystem may provide more support for networks and swarms of UAVs in the future [22].

Since these GCS applications are open-source, software definitions of the aforementioned policy recommendations could feasibly be implemented. For example, if packet loss moving average falls below the defined threshold for BVLOS missions, then an RTL command could be sent to the UAV, although a more pragmatic approach should be taken. Perhaps in this case a maximum BVLOS distance should be set during mission, consequently updating the flight path in accordance with this new limitation. Current development of the MAVLink protocol aims to introduce a wide variety of features, one such message type, MISSION_CHANGED, would support this type of scenario [18]. This message is sent to the GCS by the UAV if the companion computer alters the current mission, an improvement upon the current exchange, requiring a complete transaction of the mission's data in it's entirety. The following are some suggestions for features that could be implemented in open-source GCS software to assist pilots in abiding by existing and future regulations:

- User defined control transmission period of each MAVLink message in a telemetry link. Presets could be used for a regulatory minimum benchmark, recommended rates, and high bandwidth rates.
- BVLOS information and options such as UAV roles (relay or end-point), network capacity, and latency.
- Moving average for packet loss and packet loss threshold to prevent unreliable UAV operations, VLOS or BVLOS.
- Boundaries and restrictions based on airspace classification and authorisation. Authentication could be used within GCS software to verify the pilot/supervisor attempting to fly in restricted or other publicly off-limits classification of airspace.

Each of the aforementioned suggestions for potential GCS features could feasibly be implemented in any open-source option. This would not only help prospective pilots fly UAVs in accordance with regulations, but it would also inspire closer collaboration between policy makers and the engineers who develop such features.

Chapter 7

Conclusion

This research set out to enhance the way Antarctic snow depth data is collected with the snow radar, by maximising the range data collection UAVs can operate by maintaining reliable telemetry in BVLOS spatial conditions. Data collection UAVs were constrained to an altitude of 5 m to 15 m due to the snow radar's limitations, and the GCS being situated at 15 m altitude while on an icebreaker. Under these spatial conditions VLOS is lost at approximately 10 km due to the Earth's curvature and surface roughness, hence the aim to maintain reliable telemetry at up to 10 km.

Research was undertaken to establish a fundamental understanding of UAV technology. With a thorough understanding of the modern open-source UAV ecosystem gained, a PoC BVLOS modular system was designed, utilising DigiMesh capable radios as a replacement to traditional PtP SiK telemetry radios. Replacing the PtP topology with a self-forming, self-healing, and automatically routed mesh network paved the way for a benchmark design that allowed for multiple UAVs to maintain telemetry with the GCS via a relay UAV. This was proved in both field and lab trials, although trials unveiled a noticeable weakness of DigiMesh – a lower than anticipated practical network capacity. Data-sheet specifications for network capacity stated 91 kbps, yet attempts to achieve this throughput resulted in a practical maximum of 41 kbps. Combining this with the telemetry rate constraints described in Equation 3.1 for each UAV, implied that the network bottle-neck between GCS and relay would limit telemetry rates to 13.67 kbps with only one data collector UAV on the network. Consequently, a hybrid network topology was developed that utilised IEEE 802.11n capable URM2 hardware for wireless communications between the GCS and relay UAV, eliminating the network bottle-neck. The telemetry rate constraint was then redefined with Equation 5.1, meaning three data collector UAVs could operate in tandem, each with a telemetry rate of 13.67 kbps.

Following the testing and verification of the PoC design, an analysis was carried out on physical layer communications to determine theoretical operational ranges. This involved simulating FSPL and the two-ray channel model for both the 915 MHz XBee, and the 2.4 GHz URM2 radio hardware. A directional antenna was assumed to be part of the GCS configuration,

thereby mitigating the secondary path of the two-ray model. Under simulation conditions the maximum horizontal range between GCS and relay UAV for each 802.11n MCS 0-15 was found. Optimising for range and network capacity with one spatial stream suggests MCS 0 and MCS 7 respectively, as anticipated – MCS 0 with a 6.5 Mbps network capacity and 44.40 km range, and MCS 7 with a 65 Mbps network capacity and 1.75 km range. Maximising the link capacity while attempting to maintain the range goal of 10 km would suggest that MCS 4 is the optimal choice, with a simulated range of 9.94 km (factoring in a +2 dB receiver tolerance) and capacity of 39 Mbps. Spatial boundaries imposed by using a directional antenna to mitigate the two-ray model were also mathematically described for a fixed antenna and antenna tracking scenarios.

Under simulation conditions, it was also suggested that the spatial position of the relay UAV could be locally optimised to improve the signal strength between relay UAV and data collector UAV. This is a result of interference of the LOS and non-LOS paths the wireless signal takes. The severity of this effect was highlighted when comparing spatial positions of the relay UAV that result in *deconstructive* and *constructive* interference. Received signal power at 2.5 km was simulated to be -80.6 dBm, less than the received signal power of -80.2 dBm simulated at 11.9 km. However, the theory that the spatial position of the relay UAV can be locally optimised relies on two primary assumptions: that the snow/ice cover is a perfectly level reflective surface, and that permittivity of ice and snow is constant. Either of the two assumptions not holding will make the interference pattern less predictable, and no longer a simple function of relay height and horizontal distance alone. It has been recommended that further analysis is carried out in Antarctica to determine the accuracy of these assumptions, and therefore, the accuracy of the two-ray model simulations described in this thesis. The data-collection foundation for such an analysis has been described using SDRs, with a algorithm for CFO correction tested and implemented to improve data quality.

Striving improve upon the PoC benchmark established, a review of open wireless networking standards was carried out. It was determined that IEEE 802.11ah was optimal among the IEEE 802.11 family of standards, as it enables an IP addressable WLAN, supports mesh networking, operates at S1G frequencies, and would provide improvements to network capacity when compared to DigiMesh. With a channel bandwidth of 1 MHz, maximising range and minimising network capacity with MCS 10 results in a maximum network capacity of 150 kbps. For one spatial stream, the network capacity can be increased to 86.7 Mbps with a 16 MHz channel bandwidth, MCS 9 and a short guard interval. As of August 2020, the technology life-cycle of 802.11ah is in it's early stages – the standard was published in 2016, and only one vendor is providing a hardware solution on the market. Replacing the PoC wireless communications technology with 802.11ah compatible hardware will provide improvements with respect to reliability and simplicity, while also providing a larger network capacity for additional data, such as real-time snow radar measurements and video feedback.

Finally, a review of Antarctica and New Zealand UAV regulations was conducted to determine limitations that affect BVLOS data collection missions. While BVLOS flight is permissible in Antarctica with approval from COMNAP, in New Zealand BVLOS is prohibited unless operating within restricted airspace under approval of ATC, local authorities, and land owner(s). It was noted that no distinction had been made between manual flight, planned missions, or autonomous flight in any official regulations or guidelines. Manual UAV flight appeared to be the model upon which all UAV regulations have been based. In response, recommendations were put forth as to how regulations could differ between the various modes of flight, along with an explanation of how telemetry data requirements could be integrated. If modes of flight garner different regulations and telemetry data guidelines are published, then BVLOS could be considered outside of restricted airspace in New Zealand, and Antarctic guidelines could be better defined for BVLOS operations. All of the recommendations made in Chapter 6 can be reinforced with the implementation of guidelines and features into the QGroundControl application (and other open-source GCS applications). This would prove beneficial for both authorities and pilots alike.

Summarised are the key takeaways from the research undertaken in this thesis:

- Traditional PtP wireless communications was replaced using bespoke middleware and XBee S3B 900HP hardware, enabling MAVLink telemetry over DigiMesh, a self-forming, self-healing, mesh networking protocol. This PoC system design enables collection of snow depth data in Antarctica during BVLOS missions, and therefore, at far greater distances than was previously possible.

- Simulations of the two-ray channel model indicate that local spatial optimisation of the relay UAV can be achieved, prompting further investigation to determine how similar this model is to the Ross Sea wireless channel – the data collection basis for which has been defined.

- IEEE 802.11ah is the optimal open networking standard for BVLOS flight – providing a simple IP addressable WLAN that boasts the network capacity benefits of WiFi, the mesh capabilities of DigiMesh, and the range capabilities of S1G technology.

- Explicit distinction among manual flight, planned missions, and autonomous flight, will pave the way for enhancing CAA's UAV regulations in New Zealand, and COMNAP's guidelines in Antarctica. The combination of GCS technology and UAV guidelines could lead to dynamic and modern UAV regulations around flight.

Appendices

Appendix A

UAV Technology Comparison

Flight Controllers	
Name	**Features**
Pixhawk Series [55]	Open source hardware design, powerful, largest userbase, multiple versions and design variants. Created by PX4, but also supports ArduPilot though the PX4 stack. STM32F7.
Navio2	A flight controller on a Raspberry Pi 2/3/4 hat integration, custom Raspberry image from developers speed up the develpoment process. Growing user-base, offers high performance, and supports open source autopilot software.
ArduPilotMega	Open source. Based off of the Arduino Mega. Large user-base, original hardware for the ArduPilot project. Cheap hardware and relatively small form-factor.
Phenix Pro	Closed source design, supports PX4 and ArduPilot autopilot. FPGA accelerated computing for advanced sensing such as object recognition.
OcPoC-Zync Mini	Closed source design, supports PX4 and ArduPilot autopilot. FPGA accelerated computing for advanced sensing such as object recognition.
BeagleBone Blue	Multi-purpose robotics board with an abundance of I/O. Relatively low processing power.
Holybro Durandal	Closed source design, supports PX4 and ArduPilot autopilot. High processing capabilities with STM32H7.

Table A.1: Comparison of different high-performance flight controllers.

Autopilot Software Comparison	
Autopilot	**Comments**
PX4	Fixed Wing, Multi-Rotor, VTOL, Rovers. Large user base and actively maintained by hundreds of developers. Documentation is often up-to-date, clear and complete [56, 57].
ArduPilot	Fixed Wing, Multi-Rotor, VTOL, Rovers, Submarines, Antenna Trackers. Large user base and actively maintained by hundreds of developers. Areas of documentation are often outdated, unfinished or unclear [58, 59].
Cleanflight	Multi-Rotor UAV autopilot designed for racing. Used to have a large user-base. However, due to the rising popularity of ArduPilot and PX4 this autopilot is no longer actively – latest changes made to autopilot software GitHub repository were on January 5th 2019 [60].
LibrePilot	Multi-Rotor UAV. Smallest user base of the four open-source autopilots in comparison, as indicated by GitHub metrics. No longer actively maintained – latest changes made to autopilot software GitHub repository were on February 22nd 2019 [61].

Table A.2: Comparison of different autopilot software.

Ground Control Station Comparison		
Name	**Open Source**	**Operating System(s)**
Mission Planner	Yes	Windows, Mac OS X
APM Planner 2.0	Yes	Windows, Mac OS X, Linux
MAVProxy	Yes	Linux
QGroundControl	Yes	Cross-Platform (including mobile)
UgCS	No	Windows, Mac OS X, Linux

Table A.3: Comparison of different GCS software applications recommended in the ArduPilot documentation [62].

Appendix B

Configuration for BVLOS Missions

All UAV companion computers are assumed to be a Linux based operating system. The GCS can run any operating system, as long as it can run QGroundControl (or equivalent alternative).

B.1 Firmware Settings & Set-Up

This section provides a guide to all firmware changes made from factory default settings. To configure a companion computer interface on TELEM2, the firmware parameters shown in Table B.1 are changed.

Parameter	Value
MAV_1_CONFIG	TELEM2
MAV_1_MODE	Onboard
SER_TEL_2_BAUD	921600

Table B.1: Reccomended Pixhawk firmware settings for interfacing with a companion computer [63].

The XBee radio settings seen in Table B.2 should be changed in XCTU.

Parameter	AT	Value
Channel Mask	CM	0x00FFFFFFFE00000000
API Mode	AP	2
Encryption Enabled	EE	0

Table B.2: Set of parameters to change in XCTU for XBee S3B radios to operate with the desired configuration. Firmware version 8075 was loaded onto XBee hardware.

For middleware scripts to run without advanced user permissions (Linux):

```
git clone https://github.com/mcdiarmid/Telemetry-via-DigiMesh.git
cd Telemetry-via-DigiMesh/
su
cp devices/rules/*.rules /etc/udev/rules.d/
udevadm control --reload && udevadm trigger
```

If URM2 or URM9 hardware is utilised for the link between GCS and relay UAV, the following configuration shown in Table B.3 describes how an Ethernet connection should be configured on a Linux computer.

Name	**Setting**
MTU	1500
IPv4 Mode	"Shared with other devices"
Address	192.168.1.100
Mask	255.255.255.0
Gateway	192.168.1.1

Table B.3: Companion computer Ethernet settings for connection to a Ubiquiti Rocket M2 or M9 device.

Table B.4 describes URM firmware settings to be changed to ensure compatability with other WiFi devices, rather than just Ubiquiti products.

Name	**Setting**
Wireless Mode	"Access Point"
Channel Width	20 MHz
airMAX Enable	False
WDS Enable	True

Table B.4: Firmware version used was XM v5.5.8.

Companion computer of UAVs, where the SSH port of 39901 has been arbitrarily selected. In this case the username of the desired account on the companion computer is "ucnavi5", but this should be changed accordingly depending on the name associated with the companion computer.

```
sudo nano /etc/ssh/sshd-config
...
Port 39901
AllowedUsers ucnavi5
```

The following execution enables a remote SSH connection to the device, which should be enabled for all companion computers for both convenience and reliability.

```
sudo systemctl --system daemon-reload
sudo systemctl enable sshd.service@ucnavi5
sudo systemctl start sshd.service@ucnavi5
```

At this stage, the only step remaining is to ensure all of the supporting software is present on the companion computer, as internet access may not be present depending on the network configuration.

```
cd ~/Documents/
git clone https://github.com/mcdiarmid/Telemetry-via-DigiMesh.git
```

B.2 Starting a BVLOS Mission

Firstly, QGroundControl should be started, with UDP/TCP connections being initiated on ports intended for each UAV. Initiate a SSH connection from the GCS computer, then start middleware processes on a data collection UAV:

```
ssh USER@IP -p PORT
ssh ucnavi5@192.168.1.205 -p 39901
...
cd ~/Documents
nohup python3 px4.py
```

Initiate a SSH connection from the GCS computer, then start middleware processes on a relay UAV:

```
ssh USER@IP -p PORT
ssh ucnavi5@192.168.1.205 -p 39901
...
cd ~/Documents
nohup python3 px4.py --ssh
python3 gcs.py
```

nohup is not specified for gcs.py, as typically the SSH connection with the relay UAV is maintained throughout the mission, and therefore, it does not need to run as a background process. This also enables the pilot to view any real-time logging information that may be of interest.

Chapter 8

Bibliography

[1] R.K. Pachauri and L.A. Meyer (eds.), "Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change," Geneva, Switzerland, 2014.

[2] T. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P. e. Midgley, "Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change." Cambridge, United Kingdom and New York, NY, USA, 2013.

[3] A. Tan, K. Eccleston, I. Platt, I. Woodhead, W. Rack, and J. McCulloch, "The design of a uav mounted snow depth radar: Results of measurements on antarctic sea ice," in *2017 IEEE Conference on Antenna Measurements Applications (CAMA)*, 2017, pp. 316–319.

[4] D.N. Thomas (ed.), "Sea Ice (3rd edition)," Chichester, UK, 2017.

[5] A. E. . Tan, K. W. Eccleston, I. Platt, I. Woodhead, W. Rack, and J. McCulloch, "Microwave measurements of snow over sea-ice in antarctica," in *2018 12th International Conference on Electromagnetic Wave Interaction with Water and Moist Substances (ISEMA)*, June 2018, pp. 1–9.

[6] A. E. Tan, J. McCulloch, W. Rack, I. Platt, and I. Woodhead, "Radar measurements of snow depth over sea ice on an unmanned aerial vehicle," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–8, 2020.

[7] S. Rosati, K. Krużelecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic routing for flying ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1690–1700, March 2016.

[8] S. Choi, J. Park, and J. Kim, "A networking framework for multiple-heterogeneous unmanned vehicles in fanets," in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2019, pp. 13–15.

[9] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, "Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives," *IEEE Access*, vol. 7, pp. 81 057–81 105, 2019.

[10] The Secretariat of the Antarctic Treaty, "The Antarctic Treaty," 2020, https://ats.aq/e/antarctictreaty.html last accessed on 2020-03-24.

[11] Antarctica New Zealand, "Antarctica New Zealand — About Us," 2020, https://www.antarcticanz.govt.nz/about-us last accessed on 2020-05-20.

[12] The Secretariat of the Antarctic Treaty, "Environmental Guidelines for operation of Remotely Piloted Aircraft Systems (RPAS) in Antarctica (v 1.1)," 2020, https://ats.aq/devAS/Meetings/Measure/679 last accessed on 2020-03-24.

[13] COMNAP RPAS Working Group, "Antarctic Remotely Piloted Aircraft Systems (RPAS) Operator's Handbook," 2019, http://www.comnap.aq/documents/COMNAP-RPAS-Handbook-25-October-2019-PDF-Format.pdf last accessed on 2020-06-30.

[14] J. Luomala and I. Hakala, "Effects of temperature and humidity on radio signal strength in outdoor wireless sensor networks," in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2015, pp. 1247–1255.

[15] A. Goldsmith, *Wireless Communications*. Stanford University, 2004. [Online]. Available: web.cs.ucdavis.edu/~liu/289I/Material/book-goldsmith.pdf

[16] S. Evans, "Dielectric properties of ice and snow–a review," *Journal of Glaciology*, vol. 5, no. 42, p. 773–792, 1965.

[17] C. Matzler, "Microwave permittivity of dry snow," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 2, pp. 573–581, 1996.

[18] MAVLink Developer Guide. [Online]. Available: https://mavlink.io/en/

[19] L. Meier. How I accidentally created the most used standards in the drone industry. [Online]. Available: https://auterion.com/the-history-of-pixhawk/

[20] Serialization. [Online]. Available: https://mavlink.io/en/guide/serialization.html

[21] Message signing. [Online]. Available: https://mavlink.io/en/guide/message_signing.html

[22] L. Meier. MAVLink3 Update - Dr. Lorenz Meier - PX4 Developer Summit 2019. Youtube. [Online]. Available: https://www.youtube.com/watch?v=-ftop8lvSPE

[23] C. A. A. of New Zealand. (2018) Part 101: Gyrogliders and parasails, unmanned aircraft (including balloons), kites, and rockets – operating rules. [Online]. Available: https://www.aviation.govt.nz/assets/rules/consolidations/Part_101_Consolidation.pdf

[24] ——. (2020) Who we are. [Online]. Available: https://www.aviation.govt.nz/about-us/who-we-are/

[25] ——. (2015) Part 102: Unmanned aircraft operator certification. [Online]. Available: https://www.aviation.govt.nz/assets/rules/consolidations/Part_102_Consolidation.pdf

[26] ——. (2008) Part 71: Designation and classification of airspace. [Online]. Available: https://www.aviation.govt.nz/assets/rules/consolidations/Part_071_Consolidation.pdf

[27] S. Haykin, *Communications Systems*. New York: John Wiley & Sons, Inc., 2001.

[28] B. P. Lathi and Z. Ding, *Modern Digital and Analog Communications Systems*. New York: Oxford University Press, 2009.

[29] Line of sight obstruction. [Online]. Available: https://s.campbellsci.com/documents/au/technical-papers/line-of-sight-obstruction.pdf

[30] P. Wireless. Definition & calculation of fresnel clearance zone. [Online]. Available: http://www.proxim.com/products/knowledge-center/calculations/calculations-fresnel-clearance-zone

[31] Fresnel zones. [Online]. Available: http://radiomobile.pe1mew.nl/?Calculations:Propagation_calculation:Fresnel_zones

[32] D. I. Inc. (2014, 5) Xbee-pro 900hp/xbee-pro xsc rf modules. [Online]. Available: https://cdn.sparkfun.com/datasheets/Wireless/Zigbee/90002173_N.pdf

[33] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1–51, 2010.

[34] "IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking," *IEEE Std 802.11s-2011 (Amendment to IEEE Std 802.11-2007 as amended by IEEE 802.11k-2008, IEEE 802.11r-2008, IEEE 802.11y-2008, IEEE 802.11w-2009, IEEE 802.11n-2009, IEEE 802.11p-2010, IEEE 802.11z-2010, IEEE 802.11v-2011, and IEEE 802.11u-2011)*, pp. 1–372, 2011.

[35] "IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016.

[36] "IEEE Standard for Information technology–Telecommunications and information exchange between systems - Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation," *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)*, pp. 1–594, 2017.

[37] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, 2016.

[38] Thread Group, "Thread overview," Tech. Rep., July 2015. [Online]. Available: https://www.threadgroup.org/support#Whitepapers

[39] Zigbee Alliance. Zigbee. [Online]. Available: https://zigbeealliance.org/solution/zigbee/

[40] Digi International Inc., "Wireless mesh networking: Zigbee vs. digimesh," Tech. Rep., 2018. [Online]. Available: https://www.digi.com/pdf/wp_zigbeevsdigimesh.pdf

[41] J. Postel, "User Datagram Protocol," The Internet Engineering Task Force (IETF), standard, August 1980. [Online]. Available: https://tools.ietf.org/html/rfc768

[42] "Transmission Control Protocol," DARPA, standard, September 1981. [Online]. Available: https://tools.ietf.org/html/rfc793

[43] ArduPilot, "Pymavlink." [Online]. Available: https://github.com/ArduPilot/pymavlink

[44] Digi International Inc., "Digi XBee Python library." [Online]. Available: https://github.com/digidotcom/xbee-python

[45] ArduPilot, "SiK - Firmware for SiLabs Si1000 - Si102x/3x ISM radios." [Online]. Available: https://github.com/ArduPilot/SiK

[46] Ubiquiti Networks, "Rocket M Datasheet," 2015.

[47] Ettus Research, "USRP™ B200mini Series Datasheet." [Online]. Available: https://www.ettus.com/wp-content/uploads/2019/01/USRP_B200mini_Data_Sheet.pdf

[48] "MAVProxy," 2020. [Online]. Available: https://github.com/ArduPilot/MAVProxy

[49] "MAVLink Router," 2020. [Online]. Available: https://github.com/intel/mavlink-router

[50] W. Rack, "Snow radar data file structure," private communication, 2020.

[51] S. Moore, "Exemption from the requirement of civil aviation rule," private communication, 2014.

[52] K. Barnsdale, "Petition for an exemption," private communication, 2014.

[53] S. Moore, private communication, 2016.

[54] "Economic Benefits to New Zealand from Beyond-Line-of-Sight Operation of UAVs," Callahan Innovation, Tech. Rep., February 2015. [Online]. Available: https://www.callaghaninnovation.govt.nz/sites/all/files/uav-benefits-study_4.pdf

[55] 3DR Pixhawk – Pixhawk 1 Flight Controller. [Online]. Available: https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk.html

[56] PX4, "Px4 drone autopilot." [Online]. Available: https://github.com/PX4/Firmware

[57] PX4 Development Guide (v1.9.0). [Online]. Available: https://dev.px4.io/v1.9.0/en/

[58] ArduPilot, "Ardupilot project." [Online]. Available: https://github.com/ArduPilot/ardupilot

[59] ArduPilot Documentation. [Online]. Available: https://ardupilot.org/ardupilot/index.html

[60] Cleanflight, "Cleanflight." [Online]. Available: https://github.com/cleanflight/cleanflight

[61] LibrePilot, "About the librepilot project." [Online]. Available: https://github.com/librepilot/LibrePilot

[62] Choosing a Ground Station. [Online]. Available: https://ardupilot.org/copter/docs/common-choosing-a-ground-station.html

[63] Companion computers. [Online]. Available: https://dev.px4.io/v1.9.0/en/companion_computer/pixhawk_companion.html